

# **6WIND vRouter - Turbo Router**

*Release 3.2.14*

**6WIND**

**Jun 22, 2022**

# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Overview</b>  | <b>1</b> |
| 1.1      | Features . . . . .   | 1        |
| 1.1.1    | Routing . . . . .  | 1        |
| 1.1.2    | LAYER 2 (Data Link Layer) and Encapsulations . . . . .       | 2        |
| 1.1.3    | IP Networking . . . . .                                      | 2        |
| 1.1.4    | IPSEC . . . . .  | 2        |
| 1.1.5    | CG-NAT (Carrier Grade Network Address Translation) . . . . . | 3        |
| 1.1.6    | Security . . . . .   | 4        |
| 1.1.7    | QoS . . . . .  | 4        |
| 1.1.8    | IP Services . . . . .  | 4        |
| 1.1.9    | Management/Monitoring . . . . .                              | 4        |
| 1.1.10   | Operations . . . . .   | 5        |
| 1.1.11   | High Availability . . . . .                                  | 5        |
| 1.2      | System Requirements . . . . .                                | 5        |
| <b>2</b> | <b>Getting Started</b>                                       | <b>7</b> |
| 2.1      | Delivery contents . . . . .                                  | 7        |
| 2.2      | Installation . . . . .                                       | 7        |
| 2.2.1    | Install on bare metal using USB stick . . . . .              | 8        |
| 2.2.2    | Install on bare metal using CDROM . . . . .                  | 12       |
| 2.2.3    | Install on bare metal using PXE . . . . .                    | 15       |
| 2.2.4    | Install as a VM (Virtual Machine) using KVM . . . . .        | 21       |
| 2.2.5    | Install as a VM using OpenStack . . . . .                    | 32       |
| 2.2.6    | Install as a VM using VMware . . . . .                       | 39       |
| 2.2.7    | Install as a VM using Proxmox VE . . . . .                   | 44       |
| 2.2.8    | Install as a VM using AWS . . . . .                          | 58       |
| 2.2.9    | Update an existing installation . . . . .                    | 63       |
| 2.3      | First configuration . . . . .                                | 66       |
| 2.3.1    | Logging in to the CLI . . . . .                              | 66       |
| 2.3.2    | Restoring from a backup file . . . . .                       | 66       |
| 2.3.3    | Day-1 configuration . . . . .                                | 67       |
| 2.3.4    | Configuring your license . . . . .                           | 70       |
| 2.3.5    | Configuring the fast path . . . . .                          | 74       |

|          |  |           |
|----------|--|-----------|
| 2.3.6    | Configuring networking . . . . .                       | 75        |
| 2.4      | Advanced Features . . . . .                            | 75        |
| 2.4.1    | Automated pre-configuration using Cloud-init . . . . . | 75        |
| <b>3</b> | <b>User Guide</b>                                      | <b>79</b> |
| 3.1      | User Guide - CLI / NETCONF . . . . .                   | 79        |
| 3.1.1    | Preface . . . . .                                      | 79        |
| 3.1.2    | Key features . . . . .                                 | 80        |
| 3.1.3    | Basics . . . . .                                       | 82        |
| 3.1.4    | System . . . . .                                       | 100       |
| 3.1.5    | Network interfaces . . . . .                           | 152       |
| 3.1.6    | IP Networking . . . . .                                | 185       |
| 3.1.7    | Routing . . . . .                                      | 217       |
| 3.1.8    | QoS . . . . .  | 466       |
| 3.1.9    | Security . . . . .                                     | 498       |
| 3.1.10   | High Availability . . . . .                            | 559       |
| 3.1.11   | Monitoring . . . . .                                   | 592       |
| 3.1.12   | Services . . . . .                                     | 606       |
| 3.1.13   | Maximum Capacity Specifications . . . . .              | 618       |
| 3.1.14   | Troubleshooting . . . . .                              | 620       |
| 3.1.15   | Automation . . . . .                                   | 628       |
| 3.2      | Command Reference . . . . .                            | 647       |
| 3.2.1    | cmd . . . . .  | 647       |
| 3.2.2    | show . . . . .   | 668       |
| 3.2.3    | flush . . . . .  | 732       |
| 3.2.4    | system . . . . .                                       | 743       |
| 3.2.5    | cloud-init . . . . .                                   | 763       |
| 3.2.6    | license . . . . .                                      | 763       |
| 3.2.7    | auth . . . . .   | 771       |
| 3.2.8    | aaa . . . . .  | 773       |
| 3.2.9    | vrf . . . . .  | 774       |
| 3.2.10   | ssh-server . . . . .                                   | 774       |
| 3.2.11   | netconf-server . . . . .                               | 776       |
| 3.2.12   | dns . . . . .  | 777       |
| 3.2.13   | lldp . . . . .   | 781       |
| 3.2.14   | kpi . . . . .  | 788       |
| 3.2.15   | telegraf . . . . .                                     | 788       |
| 3.2.16   | tracker . . . . .                                      | 790       |
| 3.2.17   | nat . . . . .  | 801       |
| 3.2.18   | cg-nat . . . . .                                       | 826       |
| 3.2.19   | ntp . . . . .  | 855       |
| 3.2.20   | firewall . . . . .                                     | 859       |
| 3.2.21   | network-port (state only) . . . . .                    | 1661      |
| 3.2.22   | interface . . . . .                                    | 1662      |
| 3.2.23   | qos . . . . .  | 1941      |
| 3.2.24   | vrrp . . . . .   | 1952      |

|          |  |             |
|----------|--|-------------|
| 3.2.25   | ike . . . . .                                    | 1975        |
| 3.2.26   | sflow . . . . .                                  | 2069        |
| 3.2.27   | snmp . . . . .                                   | 2073        |
| 3.2.28   | routing . . . . .                                | 2090        |
| 3.2.29   | DHCP . . . . .                                   | 2592        |
| 3.2.30   | fast-path . . . . .                              | 2606        |
| 3.2.31   | logging . . . . .                                | 2629        |
| 3.2.32   | high availability . . . . .                      | 2636        |
| 3.2.33   | group . . . . .                                  | 2642        |
| <b>4</b> | <b>Troubleshooting</b>                           | <b>2646</b> |
| 4.1      | Relevant Information for Bug Reporting . . . . . | 2646        |
| 4.2      | Typical issues . . . . .                         | 2647        |
| 4.2.1    | Startup Issues . . . . .                         | 2647        |
| 4.2.2    | Networking Issues . . . . .                      | 2653        |
| 4.2.3    | Performance Tuning . . . . .                     | 2657        |
| 4.2.4    | OpenStack . . . . .                              | 2658        |
| 4.2.5    | Management . . . . .                             | 2660        |
| 4.3      | Fast Path Information . . . . .                  | 2660        |
| 4.3.1    | Fast Path statistics . . . . .                   | 2660        |
| 4.3.2    | fp-cpu-usage . . . . .                           | 2661        |
| 4.3.3    | Turn Fast Path off . . . . .                     | 2662        |
| 4.4      | System Information . . . . .                     | 2662        |
| 4.4.1    | CPU Pinning for VMs . . . . .                    | 2662        |
| 4.4.2    | fp-cli dpdk-port-stats . . . . .                 | 2664        |
| 4.4.3    | lspci . . . . .                                  | 2665        |
| 4.4.4    | lstopo . . . . .                                 | 2666        |
| 4.4.5    | meminfo . . . . .                                | 2667        |
| 4.4.6    | numastat . . . . .                               | 2669        |
| 4.5      | Log Management . . . . .                         | 2670        |
| 4.5.1    | rsyslog . . . . .                                | 2670        |
| 4.5.2    | journalctl . . . . .                             | 2672        |
| 4.5.3    | fast path logs . . . . .                         | 2672        |
| 4.5.4    | fpm logs . . . . .                               | 2674        |
| 4.5.5    | cmgrd logs . . . . .                             | 2674        |
| 4.5.6    | OpenStack logs . . . . .                         | 2675        |
| 4.6      | External Tools . . . . .                         | 2678        |
| 4.6.1    | strace . . . . .                                 | 2678        |

# 1. Overview

Thank you for choosing 6WIND Turbo Router.

Turbo Router is a ready-to-use high performance software routing appliance.

Turbo Router provides Service Providers, Cloud and Content Providers, and Enterprises the best price/performance ratio when transitioning from hardware to software based appliances.

Turbo Router can be quickly installed on x86 servers in bare metal or virtual machine environments.

This document will help you get started with your new product. It provides an overview as well as detailed installation and startup instructions.

## 1.1 Features

Turbo Router offers:

- Linear performance scalability with the number of cores deployed
- Full-featured data plane networking with fast path protocols
- High performance control plane
- CLI (Command Line Interface) management
- NETCONF management
- High performance input/output (I/O) leveraging DPDK (Data Plane Development Kit) with multi-vendor NIC (Network Interface Card) support
- Bare metal and virtual environment support, including KVM, VMware and AWS

### 1.1.1 Routing

- BGP (Border Gateway Protocol), BGP4+
- OSPF (Open Shortest Path First)v2, OSPFv3
- RIP (Routing Information Protocol), RIPNG (Routing Information Protocol next generation)
- CROSS-VRF (Cross Virtual Routing and Forwarding)
- Static Routes
- Path monitoring

- ECMP (Equal Cost Multi Path)
- PBR (Policy-Based Routing)
- BFD (Bidirectional Forwarding Detection)
- MPLS (Multiprotocol Label Switching) LDP (Label Distribution Protocol) (beta)
- BGP L3VPN (Layer 3 Virtual Private Network) (beta)
- VXLAN (Virtual eXtensible Local Area Network) EVPN (Ethernet Virtual Private Network) (beta)
- Point to Multipoint GRE (Generic Routing Encapsulation) interfaces
- NHRP (Next Hop Routing Protocol)
- DMVPN (Dynamic Multipoint VPN) with IPSEC (Internet Protocol Security)

### 1.1.2 LAYER 2 (Data Link Layer) and Encapsulations

- GRE
- VLAN (Virtual Local Area Network) (802.1Q, QinQ)
- VXLAN
- LAG (Link Aggregation) (802.3ad, LACP)
- Ethernet Bridge

### 1.1.3 IP Networking

- IPv4 and IPv6
- IPv6 Autoconfiguration
- VRF (Virtual Routing and Forwarding)
- IPv4 and IPv6 Tunneling
- NAT (Network Address Translation)

### 1.1.4 IPSEC<sup>1</sup>

- IKE (Internet Key Exchange)v1, IKEv2 Pre-shared Keys or X509 Certificates
- MOBIKE
- Encryption: 3DES, AES-CBC/GCM (128, 192, 256)
- Hash: MD-5, SHA-1, SHA-2 (256, 384, 512), AES-XCBC (128)

---

<sup>1</sup> requires a Turbo IPsec Application License

- Key Management: RSA, DH MODP groups 1 (768 bits), 2 (1024 bits), 5 (1536 bits) and 14 (2048 bits), DH PFS
- High performance (AES-NI, QAT)
- Tunnel, Transport or BEET mode
- SVTI (Secure Virtual Tunnel Interface), DVTI

### 1.1.5 CG-NAT (Carrier Grade Network Address Translation)<sup>2</sup>

- NAT44 (Network Address IPv4-to-IPv4 Translation)
- NAT64 (Network Address IPv6-to-IPv4 Translation) in conjunction with DNS64 (Domain Name Service for IPv6-to-IPv4 Translation)
- Port Assignment
  - Random or parity
  - Port Block Allocation (PBA)
  - Per user/per CPE session limiter
- IP Pool Management
  - Paired pooling
  - IP pool resize
- Logging
  - Port batching
  - Syslog
- ALG support
  - ICMP, FTP, TFTP, RTSP, PPTP, SIP, H323
- Hairpinning
- Endpoint-Independent Mapping and Filtering
- Address and Port Dependent Mapping and Filtering

---

<sup>2</sup> requires a Turbo CG-NAT Application License

### 1.1.6 Security

- Access Control Lists
- Unicast Reverse Path Forwarding
- Control Plane Protection
- BGP Flowspec

### 1.1.7 QoS

- Rate limiting per interface, per VRF
- Class-based QoS
  - Classification: ToS / IP / DSCP / CoS
  - Shaping and Policing
  - Scheduling: PQ, PB-DWRR

### 1.1.8 IP Services

- DHCP (Dynamic Host Configuration Protocol) v4 client
- DHCP v4 server
- DHCP v4 relay
- DNS (Domain Name Service) client
- DNS proxy
- NTP

### 1.1.9 Management/Monitoring

- SSH (Secured SHell)v2
- CLI
- NETCONF / YANG API
- SNMP
- KPIs (Key Performance Indicators) / Telemetry (YANG-based)
- Role-Based Access Control with AAA (TACACS)
- Syslog
- 802.1ab LLDP (Link Layer Discovery Protocol)

- sFlow

### 1.1.10 Operations

- Installation: PXE, USB, ISO, QCOW2, OVA
- Update / Rollback Support
- Provisioning: cloud-init, Ansible
- Licensing: online licensing system with flexible feature and capacity enablement

### 1.1.11 High Availability

- VRRP (Virtual Router Redundancy Protocol)
- IKE/IPsec synchronization<sup>Page 2, 1</sup>

## 1.2 System Requirements

- Bare metal or VM (KVM, VMware, AWS)
- Virtio vNIC, VMXNET3, PCI (Peripheral Component Interconnect) passthrough and SR-IOV (Single Root I/O Virtualization)
- Supported processors
  - Intel Xeon E5-1600/2600/4600 v2 family (Ivy Bridge EP)
  - Intel Xeon E5-1600/2600/4600 v3 family (Haswell EP)
  - Intel Xeon E5-1600/2600/4600 v4 family (Broadwell EP)
  - Intel Xeon E7-2800/4800 v2 family (Ivy Bridge EX)
  - Intel Xeon E7-2800/4800 v3 family (Haswell EX)
  - Intel Xeon E7-4800/8800 v4 family (Broadwell)
  - Intel Xeon Platinum/Gold/Silver/Bronze family (Skylake)
  - Intel Atom C3000 family (Denverton)
  - Intel Xeon D family
- Supported Ethernet NICs
  - Intel 1G 82575, 82576, 82580, I210, I211, I350, I354 (igb)
  - Intel 10G 82598, 82599, X520, X540 (ixgbe)
  - Intel 10G/40G X710, XL710, XXV710 (i40e)

- Mellanox 10G/25G/40G/50G/100G Connect-X 4/5 (mlx5)
- Broadcom NetExtreme E-Series (bnxt)
- Memory footprint (RAM): Turbo Router requires at least 2GB of RAM. Default capabilities are automatically adjusted to the amount of RAM available.

Turbo Router requires 8G of RAM to achieve the following capabilities:

|   |   |
|---|---|
| VRs (Virtual Routers)                       | 32  |
| Routes                                      | 1000000   |
| Neighbors                                   | 100000  |
| PBR rules                                   | 4096  |
| Netfilter rules                             | 10000   |
| Netfilter contracks                         | 262144  |
| Netfilter ebttables                         | 10000   |
| Netfilter ipset                             | 64 ipsets per VR (Virtual Router), 2048 entries per ipset |
| VXLAN interfaces                            | 512   |
| IPsec tunnels <sup>Page 2, 1</sup>          | 100000  |
| CG-NAT Max contracks <sup>Page 3, 2</sup>   | 4M  |
| CG-NAT Max NAT entries <sup>Page 3, 2</sup> | 4M  |
| CG-NAT Max cpe (users) <sup>Page 3, 2</sup> | 20K   |
| CG-NAT Max blocks <sup>Page 3, 2</sup>      | 80K   |

---

**Note:** Some of these numbers (CG-NAT) are empirical. They may have to be tuned according to your use case.

---

**See also:**

*Fast path limits configuration* to tune these capabilities.

- CPU: Turbo Router requires at least 2 CPU cores.
- Storage: Turbo Router requires at least 1GB of storage space; 8GB are recommended to manage several images and store configuration and log files.

## 2. Getting Started

This section explains how to install, update and configure Turbo Router.

### 2.1 Delivery contents

The Turbo Router delivery contains:

- `bin/`  
Turbo Router images in various formats, as described in the *Installation* section.
- `.md5` files to check integrity of deliverables

### 2.2 Installation

This section explains how to install a Turbo Router appliance.

Turbo Router is provided in several flavors matching different installation methods.

| Method  | Flavor                        |
|---|-------------------------------|
| <i>Install on bare metal using USB stick</i>          | <code>img.gz</code>           |
| <i>Install on bare metal using CDROM or using PXE</i> | <code>iso</code>              |
| <i>Install as a VM using KVM or using OpenStack</i>   | <code>qcow2</code>            |
| <i>Install as a VM using VMware</i>                   | <code>ova</code>              |
| <i>Install as a VM using Proxmox VE</i>               | <code>iso</code>              |
| <i>Install as a VM using AWS</i>                      | <code>ami</code> <sup>1</sup> |
| <i>Update an existing installation</i>                | <code>update</code>           |

After you have installed Turbo Router following one of the methods above, jump to the *First configuration* section.

<sup>1</sup> Contact your customer support representative to get access to a private ami.

## 2.2.1 Install on bare metal using USB stick

This chapter explains how to try Turbo Router on a physical machine, and install it, using a USB stick.

The first thing to do is to *create the USB stick*.

When it is done, you can either:

- *Test Turbo Router* without changing anything on your machine
- *Install Turbo Router* on a local disk

### Create the USB stick

You will need a 2GB USB stick at least, and a Linux system. The data on the USB stick will be lost in the process.

We need to find which device will be associated to the USB stick in the Linux system. One way to do it is to use `lsblk`.

Before plugging the USB stick, run:

```
$ lsblk | grep disk
sda      8:0      0 698.7G  0 disk
sdb      8:16     0 931.5G  0 disk
```

Then plug the USB stick. A new device should appear:

```
$ lsblk | grep disk
sda      8:0      0 698.7G  0 disk
sdb      8:16     0 931.5G  0 disk
sdc      8:32     1  14.4G  0 disk
```

In our case, `sdc` is the device associated to the USB stick.

**Warning:** Please carefully check the device associated to your USB stick, or you could wipe your local drive in the next step.

---

**Note:** Make sure that your usb device was not auto-mounted before performing the next steps with `mount -l`. If it was, use the `umount` command to unmount each mounted partition.

---

Once you know this device, you can put the turbo `img.gz` file on the Linux system, unzip it and put it on the USB device.

```
# gunzip 6wind-vrouter-tr-ae-*.img.gz
# dd if=6wind-vrouter-tr-ae-*.img of=/dev/sdc bs=8M
```

---

**Note:** These two commands will take several minutes to complete. The progress of the `dd` command can be checked by doing `kill -USR1 $(pgrep ^dd)`.

---

## Test Turbo Router

You will need physical access to the machine, and a keyboard and screen attached to it to complete these steps. Alternately, you may access the machine using its first serial port.

Once the USB stick is ready, it has to be plugged in the machine on which you want to test Turbo Router.

**Warning:** Please make sure that there is no other Turbo Router live CDROM or live USB inserted in this machine. Otherwise the system might fail to boot properly.

Then, you should go in the BIOS setup, select the USB stick as first boot device, save the configuration, and reboot. After some time, you should get an output similar to the following on screen.

```
GNU GRUB  version 2.02
+-----+
|*Turbo Router - X.Y.Z|
|                       |
|                       |
|                       |
|                       |
|                       |
|                       |
|                       |
|                       |
+-----+

Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands
before booting or `c' for a command-line.
The highlighted entry will be executed automatically in 9s.
```

After 10 seconds, or if you type on the Enter key, the boot will start. You should get the following output.

```

  _____
 /  ^  \    /  /  _  |  \  |  |  _  \  _  _  |  _  \  _  _  _  |  _  \  _  _  _  |  _  \  _  _  _  |
|  '  \  \  ^  /  /  |  |  |  \  |  |  |  |  \  \  /  /  |  \  /  _  \  |  |  |  |  _  /  _  \  '  _  |
|  (  )  \  v  v  /  |  |  |  \  |  |  |  |  \  v  /  |  _  <  (  )  |  |  |  |  |  _  /  |
 \  _  /  \  ^  \  /  |  _  |  |  \  |  _  /  \  /  |  |  \  \  _  /  \  _  ,  -  \  \  _  _  |  |  |
vrouter login:

```

You are ready to test the software. Your data will persist on the USB stick.

The next step is to perform your *first configuration*.

## Install Turbo Router

Once you have tried Turbo Router, you can install it on your machine.

It can be done from the CLI, using the `system-image` command.

But first, you need to know on which device Turbo Router should be installed. To do so, log in as admin, password admin, and at the prompt, do:

```

vrouter> show state system linux disk-usage
disk-usage sda
    total 15461882265
    ..
disk-usage sdb
    total 1000190509056
    ..

```

sda is the USB stick, which we do not want to break. It is the first detected device, and its size is small (14.4G in our case).

sdb is the device we are looking for, it is much bigger. We will install Turbo Router on sdb in our example. The data on sdb will be lost in the process.

**Warning:** Please carefully check the device associated to the disk you want to use, or you could wipe the wrong drive in the next step.

---

**Note:** Please make sure to select this disk as boot device after installation.

---

Now, do:

```

vrouter> cmd system-image install-on-disk sdb

```

This command will install Turbo Router on /dev/sdb. The relevant configuration files will be copied from the USB stick to the local drive. At the end of the installation, you can reboot and remove the USB stick.

---

**Note:** To restore from a backup file, add `backup-url <url>` to the previous command. This will restore your configurations, private keys, certificates and licenses.

The backup file must have been generated on the same or previous minor version (e.g. a backup from 3.0.1 can be restored on 3.0.x or 3.1.x).

---

You will then get the familiar GRUB screen that you got when you were testing the software, and after some time, the login screen.

```

GNU GRUB  version 2.02

+-----+
|*Turbo Router - X.Y.Z|
|                       |
|                       |
|                       |
|                       |
|                       |
|                       |
+-----+

Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands
before booting or `c' for a command-line.
The highlighted entry will be executed automatically in 9s.

(...)

  _____
 /  ^  \    /  /  _  | \  | |  _  \  _  _  |  _  \  _  _  _  |  |  _  _  _  _
| ' _ \ \ ^ / / | | | \ | | | | \ \ / / | ) / _ \ | | | | _ / _ \ ' _ |
| ( ) \ V V / | | | \ | | | | \ V / | _ < ( ) | | | | | | _ / |
 \__ / \ ^ / | _ | | \ | _ _ /   \ / | | \ \ _ / \ _ , | \ \ _ _ | |

vrouter login:

```

The next step is to perform your *first configuration*.

## 2.2.2 Install on bare metal using CDROM

This chapter explains how to try Turbo Router on a physical machine, and install it, using a CDROM drive either physical or virtual.

If your server has a physical CD/DVD drive, you first need to burn the `iso` file on a blank CD or DVD. If it provides a virtual CDROM feature, simply use the `iso` file as input.

When you're done, you can either:

- *Test Turbo Router* without changing anything on your machine
- *Install Turbo Router* on a local disk

### Test Turbo Router

You will need physical access to the machine, and a keyboard and screen attached to it to complete these steps. Alternately, you may access the machine using its first serial port.

Once your CDROM setup is ready, it has to be inserted in the machine on which you want to test Turbo Router.

**Warning:** Please make sure that there is no other Turbo Router live CDROM or live USB inserted in this machine. Otherwise the system might fail to boot properly.

Then, you should go in the BIOS setup, select the CDROM drive as first boot device, save the configuration, and reboot.

After some time, you should get an output similar to the following on screen.

```
GNU GRUB  version 2.02
+-----+
|*Turbo Router - X.Y.Z|
|                       |
|                       |
|                       |
|                       |
|                       |
|                       |
|                       |
|                       |
|                       |
+-----+
```

(continues on next page)

(continued from previous page)

Use the ^ and v keys to select which entry is highlighted.  
 Press enter to boot the selected OS, `e` to edit the commands before booting or `c` for a command-line.  
 The highlighted entry will be executed automatically in 9s.

After 10 seconds, or if you type on the Enter key, the boot will start. You should get the following output.

```

  ____
 /  \ \      /  /  _ | \ | |  _ \  _  _ |  _ \  _  _  _ |  _ |  _  _  _  _
 | ' _ \ \ \ / / / | | | \ | | | | \ \ / / |_) / _ \ | | | |  _ / _ \ ' _ |
 | ( ) \ V  V /  | | | \ | | | | \ V / |  < ( ) | | | | |  _ / |
 \__ / \ \ \ /  |__| | | \ |__ /   \ / | | \ \ \ / \ \ , - | \ \ \ | | |
vrouter login:

```

You are ready to test the software. Your data will not persist after a reboot.

The next step is to perform your *first configuration*.

### Install Turbo Router

Once you have tried Turbo Router, you can install it on your machine.

It can be done from the CLI, using the `system-image` command.

But first, you need to know on which device Turbo Router should be installed. To do so, log in as admin, password admin, and at the prompt, do:

```

vrouter> show state system linux disk-usage
disk-usage sda
  total 1000190509056
  ..

```

sda is the device we are looking for. We will install Turbo Router on sda in our example. The data on sda will be lost in the process.

**Warning:** Please carefully check the device associated to the disk you want to use, or you could wipe the wrong drive in the next step.

**Note:** Please make sure to select this disk as boot device after installation.

Then launch the installation on sda.



(continued from previous page)

```
vrouter login:
```

The next step is to perform your *first configuration*.

## 2.2.3 Install on bare metal using PXE

This chapter explains how to deploy Turbo Router on a set of physical machines via PXE and make them available for remote access (e.g. SSH, Ansible, etc.).

The procedure relies on the Turbo Router `iso` file and requires a deployment infrastructure enabling PXE by providing DHCP, TFTP, DNS and HTTP services.

- *Install a PXE server*
- *Configure the PXE server*
- *Deploy Turbo Router on the target*

### Install a PXE server

This section describes the installation and the configuration of the required packages on an Ubuntu 16.04 server to provide DHCP, DNS, TFTP and HTTP services for PXE.

First, install the required packages as root:

```
apt-get update
apt-get install -y apache2 apache2-bin apache2-data apache2-utils dnsmasq \
dnsmasq-base grub-common grub-pc grub-pc-bin grub2-common
```

Configure the network interface that will answer DHCP requests in `/etc/network/interfaces` (adapt address and netmask to your environment):

```
[...]
auto eth1
iface eth1 inet static
    address 192.168.235.1
    netmask 255.255.255.0
[...]
```

And bring this interface up:

```
ifup eth1
```

Then, configure `dnsmasq` to provide DHCP, DNS and TFTP services for your network. Edit `/etc/dnsmasq.conf` with the following contents:

```
vi /etc/dnsmasq.conf
# Listening interfaces
interface=eth1
# DNS configuration
bogus-priv
no-hosts
domain=pxeserver.com
# DHCP configuration
dhcp-range=192.168.235.10,192.168.235.150,12h
dhcp-host=14:18:77:66:c7:23,host1,192.168.235.13,infinite
dhcp-host=52:54:00:12:34:57,host2,192.168.235.36
dhcp-boot=boot/grub/i386-pc/core.0
# TFTP configuration
enable-tftp
tftp-root=/var/lib/tftpboot
```

**See also:**

the `dnsmasq` man page (<http://manpages.ubuntu.com/manpages/bionic/man8/dnsmasq.8.html>) for more information about the configuration options.

Create the root directory for the TFTP server:

```
grub-mknetdir --net-directory=/var/lib/tftpboot
```

---

**Note:** in the rest of this document, `/var/lib/tftpboot` will be referred to as `$TFTP_DIR`.

---

Then, restart the `dnsmasq` service:

```
systemctl restart dnsmasq
```

Finally, configure the `apache2` HTTP server.

Edit the default configuration file in `/etc/apache2/sites-available/000-default.conf`:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    RewriteEngine On
    RewriteRule ^/cloud-init/(.*) /%{REMOTE_ADDR}/$1
</VirtualHost>
```

---

**Note:** in the rest of this document, `/var/www/html` will be referred to as `$HTTP_DIR`.

---

Enable the URL rewriting module:

```
a2enmod rewrite
```

Then, restart apache2:

```
systemctl restart apache2
```

You are now ready to configure the PXE server.

## Configure the PXE server

This section describes how to use the Turbo Router deliverables and the PXE server together to finalize the PXE infrastructure.

First, copy the Turbo Router deliverables into the proper directories.

Kernel and filesystem of the installer:

```
cp vmlinuz initrd.img $TFTP_DIR/
```

Turbo Router ISO image:

```
cp 6wind-vrouter-tr-ae-*.iso $HTTP_DIR/turbo-router.iso
```

Then, create the `$TFTP_DIR/boot/grub/grub.cfg` file that will be provided to PXE targets:

```
vi /var/lib/tftpboot/boot/grub/grub.cfg
set timeout=5
menuentry 'Turbo Router network installer' {
    set root='(pxe)'
    set kernel_image="/vmlinuz"
    set ramdisk="/initrd.img"
    set boot_opts="ro rd.debug console=tty1 fsck.mode=skip"
    set boot_opts="$boot_opts BOOTIF=01-$net_default_mac boot=live nonnetworking_
↪console=ttyS0,115200n8 splash"
    set boot_opts="$boot_opts live-media-path=/iso/"
    set boot_opts="$boot_opts persistence persistence-storage=directory,filesystem_
↪persistence-path=/iso/ persistence-label=ramdisk_Data"
    set boot_opts="$boot_opts fetch=http://$pxe_default_server/turbo-router.iso_
↪ds=nocloud-net;s=http://$pxe_default_server/cloud-init/"
    echo "Boot options: $boot_opts"
    echo "Loading kernel image $kernel_image ..."
```

(continues on next page)

(continued from previous page)

```

linux $kernel_image $boot_opts
initrd $ramdisk
}

```

**See also:**

the GRUB documentation (<https://www.gnu.org/software/grub/manual/grub/grub.html>) for more information about the configuration options.

Next, prepare per-target cloud-init configurations. The previous configuration will make targets retrieve their respective cloud-init meta-data and user-data configurations from \$HTTP\_DIR/\$CLIENT\_IP, \$CLIENT\_IP being the address assigned to the host by DHCP.

```

mkdir $HTTP_DIR/$CLIENT_IP/

cat > $HTTP_DIR/$CLIENT_IP/meta-data <<EOF
instance-id: host1
local-hostname: host1
EOF

cat > $HTTP_DIR/$CLIENT_IP/user-data <<EOF
#cloud-config

users:
- name: root
  lock_password: true
  ssh_authorized_keys:
  - ecdsa-sha2-nistp256
  ↪AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBNqR+NMuQUywXp5+uqSc6WSFjxLRpRZoA9b7ekBeL9F

runcmd:
- [/usr/bin/wget, 'http://192.168.235.1/cloud-init/vrouter.startup', -O, /run/vrouter.
  ↪startup]
- [/usr/bin/sysrepcfg, -m, vrouter, -d, startup, -f, json, --import=/run/vrouter.
  ↪startup]
- [/usr/bin/sysrepcfg, -m, vrouter, -C, startup]
- [/usr/bin/vrouter-install.sh, -d, /dev/sda]
- [/sbin/reboot]
EOF

```

This user-data file aims at:

- disabling password access and installing an authorized public SSH key for the root user for security reasons,
- retrieving a startup Turbo Router configuration from the HTTP server (see below),
- performing the installation on the given /dev/sda disk,

- rebooting.

```

cat > $HTTP_DIR/$CLIENT_IP/vrouter.startup <<EOF
{
  "vrouter:config": {
    "vrouter-system:system": {
      "hostname": "host1",
      "vrouter-auth:auth": {
        "vrouter-embedded:default-users-enabled": false,
        "user": [
          {
            "name": "admin",
            "role": "admin",
            "authorized-key": [
              "ecdsa-sha2-nistp256_
↪AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBNqR+NMuQUywXp5+uqSc6WSFjxLRpRZoA9b7ekBeL9F
↪"
          ]
        ]
      }
    ]
  },
  "vrf": [
    {
      "name": "main",
      "vrouter-interface:interface": {
        "physical": [
          {
            "name": "mgmt0",
            "ipv4": {
              "dhcp": {
                "enabled": true
              }
            },
            "port": "pci-b0s8"
          }
        ]
      },
      "vrouter-ssh-server:ssh-server": {
        "enabled": true,
        "port": 22
      }
    }
  ]
}

```

(continues on next page)

(continued from previous page)

```
}  
EOF
```

This startup configuration:

- sets `host1` as the hostname,
- disables Turbo Router default users and passwords for security reasons and configures an `admin` user with `admin` role and a SSH key,
- configures a management interface in `main vrf` with DHCP enabled.

You can now deploy Turbo Router.

## Deploy Turbo Router on the target

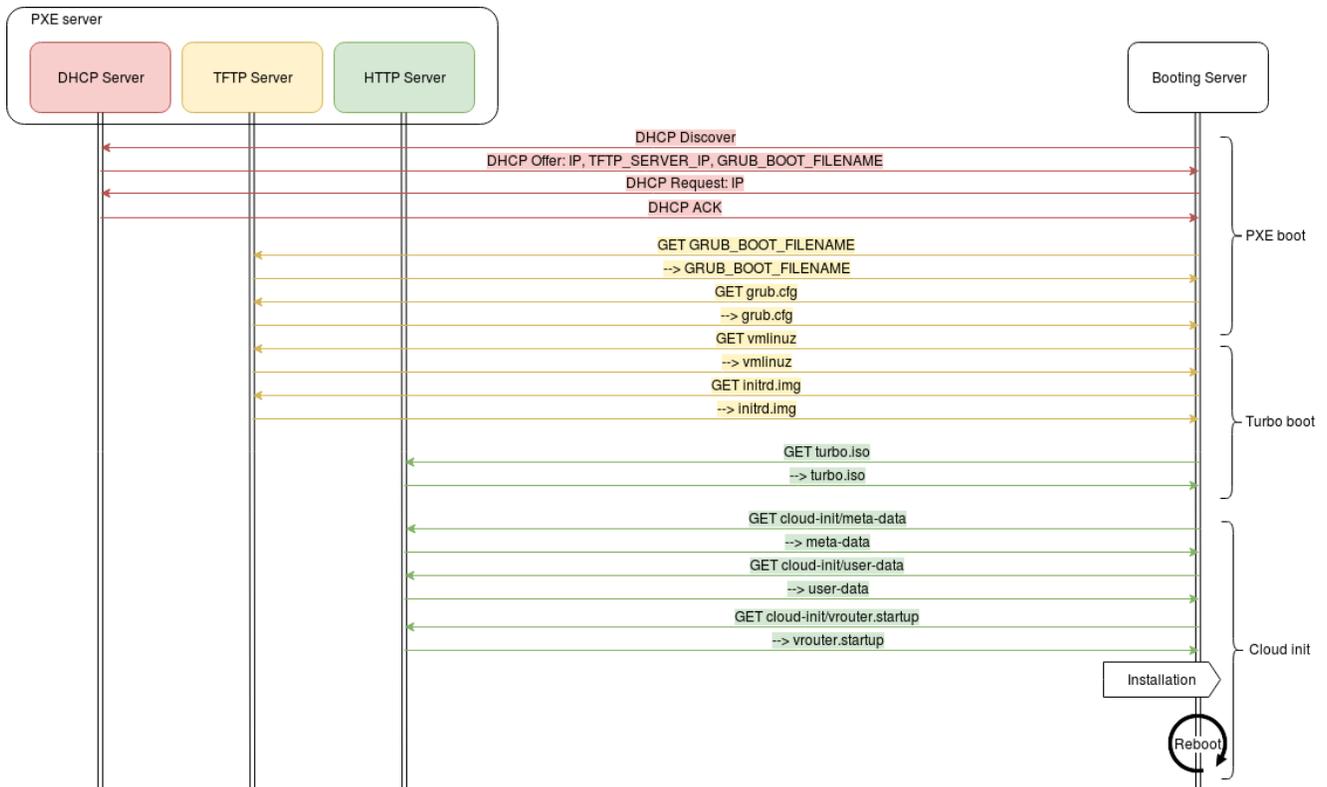
Your target must be configured to boot in Legacy BIOS mode (UEFI is not supported) and first on the hard drive selected for installation.

To start the installation, configure the target to boot using PXE. For example, using an IPMI request to perform a PXE installation on next boot only:

```
# ipmitool -I lanplus -H <BMC_IP> -U <user> chassis bootdev pxe  
# ipmitool -I lanplus -H <BMC_IP> -U <user> chassis power reset
```

On boot, the target will perform the following tasks:

- retrieve an IP address, a hostname and the TFTP server address through DHCP,
- boot the Turbo Router installer kernel and `initrd`, using the Turbo Router `iso` as root filesystem
- execute the cloud-init script to:
  - configure the root account (no password, SSH key)
  - install Turbo Router locally on the target disk device
  - install the startup configuration
  - reboot



On reboot, the normal boot sequence of the server will boot on the freshly installed hard drive, now running Turbo Router.

Thanks to the startup configuration, an IP address will be obtained on the first network interface and the console will be accessible through SSH. At this step, it is possible to automate other deployment tasks, for example using Ansible.

The next step is to perform your *first configuration*.

### 2.2.4 Install as a VM (Virtual Machine) using KVM

This chapter explains how to start a VM using KVM.

First, you should have a look at the *hypervisor prerequisites* section.

After the prerequisites are completed, you have two choices:

- a simple configuration to try Turbo Router CLI using a *VM with virtual NICs*
- a more complex configuration with good performance using a *VM with physical NICs*

---

**Note:** Most of this chapter was written for an Ubuntu 16.04 hypervisor. There should be no technical problem when using another distribution, only some commands might vary.

---

## Hypervisor prerequisites

We will not detail how to install a linux distribution here. Once it is installed, some tasks must be completed to configure the distribution into an hypervisor.

1. The `kvm` and `kvm_intel` modules have to be inserted:

```
# lsmod | grep kvm
kvm_intel          172032  0
kvm                544768  1 kvm_intel
```

2. `qemu-kvm`, `libvirt` and `virt-install` have to be installed:

```
# apt-get install -y qemu-kvm
# apt-get install -y virtinst libvirt-bin
```

or

```
# yum install -y qemu-kvm
# yum install -y virt-install libvirt
```

## VM with virtual NICs (Network Interface Cards)

In this example, the VM will have three interfaces:

- one management interface on the libvirt default virtual network using NAT forwarding,
- two data plane interfaces on top of the host's interfaces using bridged networking to connect the VM to the LAN.

### See also:

the `libvirt` networking documentation (<https://wiki.libvirt.org/page/Networking>) for more information about networking with KVM.

1. On the host, set interfaces up.

```
# ip link set eth1 up
# ip link set eth2 up
```

2. On the host, create two Linux bridges, each containing one physical interface.

```
# brctl addbr br0
# brctl addif br0 eth1
# ip link set br0 up
# brctl addbr br1
# brctl addif br1 eth2
# ip link set br1 up
```

3. To boot Turbo Router in libvirt as a guest VM, use:

```
# cp turbo-router.qcow2 /var/lib/libvirt/images/vm1.qcow2
# virt-install --name vm1 --vcpus=3,sockets=1,cores=3,threads=1 \
  --os-variant ubuntu18.04 --cpu host --network=default,model=e1000 \
  --ram 8192 --noautoconsole --import \
  --disk /var/lib/libvirt/images/vm1.qcow2,device=disk,bus=virtio \
  --network bridge=br0,model=e1000 --network bridge=br1,model=e1000
```

4. Connect to the VM:

```
# virsh console vm1
Connected to domain vm1
Escape character is ^]

      _____
     /  ^  \   /  /  _  | \  | |  _  \  __  _  |  _  \  __  _  |  _  \  __  _  |
    | ' _ \ \ ^ / /  | | | \  | | | | \ \ / / | ) /  _  \  | | | |  _  \  ' _ |
    | ( ) \ v  v /  | | | \  | | | | \ v / |  < ( ) | | | |  _  \  |
    \___/ \ ^ / \  | | | | \  | | | | \ / | | | \ \___/ \___, - | \ \___/ | |

vrouter login: admin
```

The next step is to perform your *first configuration*.

## VM with physical NICs

This section details how to start Turbo Router with dedicated physical NICs.

Using dedicated NICs requires some work which is detailed in *Hypervisor mandatory prerequisites*.

Once the hypervisor is configured properly, two technologies are available:

- whole NICs are dedicated to Turbo Router, see *Passthrough mode*, simpler configuration, but only one VM can use each NIC
- portions of NICs are dedicated to Turbo Router, see *SR-IOV mode*, to have more VMs (Virtual Machines) running on the hypervisor

For production setups, you might want to consider checking *Optimize performance in virtual environment* to get the best performance.

## Hypervisor mandatory prerequisites

### enable Intel VT-d

Intel VT-d stands for “Intel Virtualization Technology for Directed I/O”. It is needed to give a physical NIC to a VM. To enable it:

- it usually has to be enabled from the BIOS. The name of this feature can differ from one hardware to the other, we advise you to check your hardware documentation to enable it.
- it has to be enabled also in the kernel, by adding `intel_iommu=on iommu=pt` in the kernel command line.

To do so, run:

```
# echo 'GRUB_CMDLINE_LINUX="$GRUB_CMDLINE_LINUX intel_iommu=on iommu=pt"' \
>> /etc/default/grub
# update-grub2
# reboot
```

You can check the boot logs at next boot to verify that Intel VT-d is properly enabled.

```
# dmesg |grep "Intel(R) Virtualization Technology for Directed I/O"
[ 1.391229] DMAR: Intel(R) Virtualization Technology for Directed I/O
```

### hugepages

For performance reasons, the memory used by the VMs that will harbor Turbo Router must be reserved in hugepages.

---

**Note:** A hugepage is a page that addresses more memory than the usual 4KB. Accessing a hugepage is more efficient than accessing a regular memory page. Its default size is 2MB.

---

`hugeadm` can be used to managed hugepages. It is part of the `hugepages` deb package and `libhugetlbfs-utils` rpm package.

To see if your system already has hugepages available, and which sizes are supported, do:

```
# hugeadm --pool-list
      Size  Minimum  Current  Maximum  Default
  2097152         0         0         0         *
1073741824       0         0         0
```

On this system, 2MB and 1GB pages are supported.

If your hardware has several sockets, for performance reason, the memory should be allocated on the same node as the interfaces that will be dedicated to the Turbo Router VM.

1. `numactl` can show which memory node should be chosen for a particular interface. Look for `membind` in the following command output. This NIC is on memory node 1.

```
# numactl -m netdev:ens4f0 --show
policy: bind
preferred node: 1
physcpubind: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
↳26 27 28 29 30 31 32 33 34 35 36 37 38 39
cpubind: 0 1
nodebind: 0 1
membind: 1
```

2. Add 8 1GB hugepages for one Turbo Router VM to NUMA node 1. You should add this command to a custom startup script to make it persistent.

```
# echo 8 > /sys/devices/system/node/node1/hugepages/hugepages-1048576kB/nr_
↳hugepages
```

3. Check that the pages were allocated

```
# hugeadm --pool-list
      Size  Minimum  Current  Maximum  Default
      2097152      0      0      0      *
1073741824      8      8      8
```

## Passthrough mode

With this configuration, the Turbo Router VM will get dedicated interfaces.

The passthrough mode is only available if the hypervisor's hardware supports Intel VT-d, and if it is enabled (see *enable Intel VT-d*).

1. You must first find the pci id of the interfaces that will be dedicated to the Turbo Router VM.

```
# lspci |grep Ethernet
03:00.0 Ethernet controller: Intel Corporation Ethernet Connection X552/X557-AT
↳10GBASE-T
03:00.1 Ethernet controller: Intel Corporation Ethernet Connection X552/X557-AT
↳10GBASE-T
05:00.0 Ethernet controller: Intel Corporation Ethernet 10G 2P X520 Adapter (rev
↳01)
05:00.1 Ethernet controller: Intel Corporation Ethernet 10G 2P X520 Adapter (rev
↳01)
07:00.0 Ethernet controller: Intel Corporation I350 Gigabit Network Connection
↳(rev 01)
07:00.1 Ethernet controller: Intel Corporation I350 Gigabit Network Connection
↳(rev 01)
```

(continues on next page)

(continued from previous page)

2. Then use `virt-install` to spawn the VM, specifying one `host-device` argument for each device that you want to dedicate. In this example, we dedicate `03:00.0` and `03:00.1`.

```
# cp turbo-router.qcow2 /var/lib/libvirt/images/vm1.qcow2
# virt-install --name vm1 --vcpus=3,sockets=1,cores=3,threads=1 \
  --os-variant ubuntu18.04 --cpu host --network=default,model=e1000 \
  --ram 8192 --noautoconsole \
  --import --memorybacking hugepages=yes \
  --disk /var/lib/libvirt/images/vm1.qcow2,device=disk,bus=virtio \
  --host-device 03:00.0 --host-device 03:00.1
```

3. Connect to the VM:

```
# virsh console vm1
Connected to domain vm1
Escape character is ^]

      _ _ _ _ _      _ _ _ _ _      _ _ _ _ _      _ _ _ _ _      _
 /  ^  \    /  /  _  \  | | | |  \  _ _  _  \  _ _  _  \  _ _  _  \  _ _  _  \
| ' _ \ \ / \ / / | | | | \ | | | | | \ \ / / | ) / _ \ | | | | | _ / _ \ ' _ |
| ( ) \ v  v / | | | | \ | | | | | \ v / | _ < ( ) | | | | | | _ / |
 \_ _ / \ ^ \ / | _ _ | | \ | _ _ _ /    \ / | | \ \ _ _ / \ _ , - | \ \ _ _ | |

vrouter login: admin
```

To get the best performance, the VM CPUs (Central Processing Units) should be associated to physical CPUs. This is called pinning, and is described in *CPU pinning*.

The next step is to perform your *first configuration*.

## SR-IOV mode

SR-IOV enables an Ethernet port to appear as multiple, separate, physical devices called Virtual Functions (VF). You will need compatible hardware, and Intel VT-d configured. The traffic coming from each VF can not be seen by the other VFs. The performance is almost as good as the performance in passthrough mode.

Being able to split an Ethernet port can increase the VM density on the hypervisor compared to passthrough mode.

In this configuration, the Turbo Router VM will get Virtual Functions.

1. First check if the network interface that you want to use supports SR-IOV and how much VFs can be configured. Here we check for `eno1` interface.

```
# lspci -vvv -s $(ethtool -i eno1 | grep bus-info | awk -F': ' '{print $2}') |
↳grep SR-IOV
    Capabilities: [160 v1] Single Root I/O Virtualization (SR-IOV)
# lspci -vvv -s $(ethtool -i eno1 | grep bus-info | awk -F': ' '{print $2}') |
↳grep VFs
    Initial VFs: 64, Total VFs: 64, Number of VFs: 0, Function
↳Dependency Link: 00
```

- Then add VFs, and check that those VFs were created. You should add this command to a custom startup script to make it persistent.

```
# echo 2 > /sys/class/net/eno1/device/sriov_numvfs
# lspci | grep Ethernet | grep Virtual
03:10.0 Ethernet controller: Intel Corporation Ethernet Connection X552 Virtual
↳Function
03:10.2 Ethernet controller: Intel Corporation Ethernet Connection X552 Virtual
↳Function
```

- You need to set eno1 up so that VFs are properly detected in the guest VM.

```
# ip link set eno1 up
```

- Then use virt-install to spawn the VM, specifying one host-device argument for each VF that you want to give. In this example, we give the VF 03:10.0 to Turbo Router.

```
# cp turbo-router.qcow2 /var/lib/libvirt/images/vm1.qcow2
# virt-install --name vm1 --vcpus=3,sockets=1,cores=3,threads=1 \
--os-variant ubuntu18.04 --cpu host --network=default,model=e1000 \
--ram 8192 --noautoconsole --import \
--memorybacking hugepages=yes \
--disk /var/lib/libvirt/images/vm1.qcow2,device=disk,bus=virtio \
--host-device 03:10.0
```

- Connect to the VM:

```
# virsh console vm1
Connected to domain vm1
Escape character is ^]

____
/ ^ \   / / _ | \ | | _ \   _ _ | | _ \   _ _ | | _ \   _ _
| ' _ \ \ / / / | | | \ | | | \ \ / / | | / _ \ | | | _ \ ' _ |
| ( ) \ V V / | | | \ | | | \ V / | _ < ( ) | | | | | _ \ |
 \_ / \ / \ / | | | \ | | | \ / | | \ \ / \ / \_ , - | \ \ | |

vrouter login: admin
```

To get the best performance, the VM CPUs should be associated to physical CPUs. This is called pinning, and is described in *CPU pinning*.

The next step is to perform your *first configuration*.

## Optimize performance in virtual environment

To get good performance, Turbo Router needs dedicated resources. It includes:

- NICs
- CPUs

The first thing to do is to identify the resources that will be dedicated. This can be done in the *Identifying hardware resources* section.

Then, all the resources must be properly isolated, and configured, see *Isolating and configuring hardware resources*.

## Identifying hardware resources

### resource inventory

Before identifying the resources that will be dedicated to the Turbo Router VM, you need to know which NICs and CPUs are available.

It can be done using `lstopo`, which is part of the `hwloc` package.

```
# lstopo -p --merge
Machine (31GB total)
  NUMANode P#0 (16GB)
    Core P#0
      PU P#0
      PU P#20
    Core P#1
      PU P#1
      PU P#21
  (...)
    Core P#12
      PU P#9
      PU P#29
  HostBridge P#0
    PCIBridge
      PCI 1000:005b
    PCIBridge
      PCI 15b3:1013
      PCI 15b3:1013
```

(continues on next page)

(continued from previous page)

```
Net "ens1f1"
PCIBridge
  PCI 8086:1d6b
PCIBridge
  PCI 8086:1521
  Net "mgmt0"
  PCI 8086:1521
  Net "enp5s0f1"
  PCI 8086:1521
  Net "enp5s0f2"
  PCI 8086:1521
  Net "enp5s0f3"
PCIBridge
  PCI 102b:0522
  PCI 8086:1d00
  Block(Disk) "sda"
  PCI 8086:1d08
NUMANode P#1 (16GB)
  Core P#0
    PU P#10
    PU P#30
  Core P#1
    PU P#11
    PU P#31
(...)
  Core P#12
    PU P#19
    PU P#39
HostBridge P#2
  PCIBridge
    PCI 8086:1583
    PCI 8086:1583
  PCIBridge
    PCI 8086:1583
    Net "ens4f0"
    PCI 8086:1583
```

On this machine:

- logical CPUs 0 to 9, and ens1f1, mgmt0, enp5s0f1, enp5s0f2, and enp5s0f1 interfaces use NUMA node 0
- logical CPUs 10 to 19, and the ens4f0 interface use NUMA node 1

---

**Note:** NUMA (Non-uniform memory access) is a memory design, in which a hardware resource can access local

memory faster than non-local memory. The memory is organized into several NUMA nodes.

---

## resource dedication

Now that you identified your hardware, you can select which NICs and CPUs will be dedicated.

There are some constraints:

- we leave the first cpu for Linux
- CPUs must be taken on the same node as NICs
- crossing NUMA nodes costs performance, so all NICs should be taken on the same node

We recommend to start with a few CPUs, and increase when the setup is functional if needed. The example in this chapter use 3 virtual CPUs.

## Isolating and configuring hardware resources

### CPU (Central Processing Unit) isolation

The CPUs that will be dedicated to the Turbo Router VM need to be properly isolated from other processes. The more reliable way to achieve this is to isolate the CPUs at boot time, on the kernel command line, using the `isolcpus` and `rcu_nocbs` directives. For instance, adding `isolcpus=1-12,29-40 rcu_nocbs=1-12,29-40` will isolate CPUs 1 to 12 and 29 to 40. It can be added to the kernel command line by doing:

```
# echo 'GRUB_CMDLINE_LINUX="$GRUB_CMDLINE_LINUX isolcpus=1-12,29-40 rcu_nocbs=1-12,29-40"' >> /etc/default/grub
# update-grub2
# reboot
```

### CPU pinning

After the vm is created, you can use `virsh vcpupin vm1 vm-cpu cpu` to do the one-to-one pinning, using the isolated CPUs. The CPUs should be taken in the list of dedicated CPUs obtained in *Identifying hardware resources*. The setup is persistent.

For instance, the next commands will pin:

- virtual CPU 0 and CPU 2,
- virtual CPU 1 and CPU 10,
- virtual CPU 2 and CPU 4

```
# virsh vcpupin vm1 0 2
# virsh vcpupin vm1 1 10
# virsh vcpupin vm1 2 4
```

## CPU configuration

The hypervisor CPUs have to be configured for several reasons.

1. To get stable performance, it is better to disable intel\_pstate from the kernel command line:

```
# echo 'GRUB_CMDLINE_LINUX="$GRUB_CMDLINE_LINUX intel_pstate=disable"' >> /etc/
↪ default/grub
# update-grub2
# reboot
```

2. To get better performance, the CPUs should use the performance governor. You should add this command to a custom startup script to make it persistent.

```
# cpupower set -b 0
# cpupower frequency-set -g performance
```

For persistent configuration, the previous commands can be added to a custom startup script.

## IRQ (Interrupt Request) affinities configuration

Having IRQ triggered on the CPUs that are dedicated to the Turbo Router VM can result in a few packets lost from time to time. If you don't notice this problem during testing, you don't need to take care of this step.

1. To do so, first ensure that the irqbalance package is removed.

```
# apt-get remove -y irqbalance
```

or

```
# yum remove -y irqbalance
```

2. Then run this script:

```
for file in $(ls /proc/irq)
do
    if [ -f /proc/irq/$file/smp_affinity_list ]; then
        echo "irq: $file"
        echo 0-4,7 > /proc/irq/$file/smp_affinity_list
        mask=$(cat /proc/irq/$file/smp_affinity)
```

(continues on next page)

(continued from previous page)

```

fi
done
echo $mask > /proc/irq/default_smp_affinity

```

0-4,7 should be changed to the list of CPUs that are *not* dedicated to the Turbo Router VM.

For persistent configuration, the previous commands can be added to a custom startup script.

## 2.2.5 Install as a VM using OpenStack

This chapter explains how to start a Turbo Router VM using OpenStack.

It expects that you already installed an OpenStack cloud, in which you are able to spawn VMs.

You have two choices:

- a simple configuration to try Turbo Router with OpenStack using a *VM with virtual NICs*
- a more complex configuration with good performance using a *VM with physical NICs*

---

**Note:** The following commands may change depending on your OpenStack version. The important part are that the image must be imported in glance, the flavor with correct size created, and that the image and the flavor are used to start the VM. It was tested with an Ubuntu 16.04 hypervisor running the Ocata OpenStack version.

---

### VM with virtual NICs

This simple configuration imports a Turbo Router qcow2 in OpenStack, creates the right flavor, and starts a Turbo Router VM.

1. **[Controller]** Export the Turbo Router qcow2 file path:

```
# TURBO_QCOW2=/path/to/6wind-turbo-*-<arch>-<version>.qcow2
```

2. **[Controller]** Use glance to create a VM image with the Turbo Router qcow2 file:

```
# openstack image create --disk-format qcow2 --container-format bare \
    --file $TURBO_QCOW2 turbo-router
```

3. **[Controller]** Create a flavor with 8192 memory and 4 virtual CPUs.

```
# openstack flavor create --ram 8192 \
    --vcpus 4 turbo-router
```

4. **[Controller]** Create two networks:

```
# neutron net-create private1
# neutron subnet-create --name private_subnet1 private1 11.0.0.0/24
# net1=$(neutron net-show private1 | grep "\ id\ " | awk '{ print $4 }')
# neutron net-create private2
# neutron subnet-create --name private_subnet2 private2 12.0.0.0/24
# net2=$(neutron net-show private2 | grep "\ id\ " | awk '{ print $4 }')
```

5. **[Controller]** Boot the Turbo Router VM with one interface on each network:

```
# openstack server create --flavor turbo-router \
    --image turbo-router \
    --nic net-id=$net1 --nic net-id=$net2 \
    turbo-router_vm
```

6. Connect to the VM. This steps depends on your OpenStack installation. You should get:

```
(...)
```

```

  _____
 /  ^  \   /  /  _  | \  | |  _  \  __  _  |  _  \  ___  _  |  |  _  ___  ___
 | '  \  \  ^  /  /  | | |  \  | | | |  \  \  /  /  |  )  /  _  \  | | |  ___/  _  \  '  _  |
 | (  )  \  v  v  /  | | |  \  | | | |  \  v  /  |  <  (  )  |  |  |  | |  ___/  |
 \___/  \  ^  \  /  |___| |  \  |___/  \  /  |  |  \  \___/  \___,  |  \  \___|  |

```

```
vrouter login: admin
```

The next step is to perform your *first configuration*.

## VM with physical NICs

This section details how to start Turbo Router with dedicated physical NICs within OpenStack.

Using dedicated NICs requires some work on your compute node which is detailed in *Hypervisor mandatory prerequisites*.

Once the hypervisor is configured properly, two technologies are available:

- whole NICs are dedicated to Turbo Router, see *Passthrough mode*, simpler configuration, but only one VM can use each NIC
- portions of NICs are dedicated to Turbo Router, see *SR-IOV mode*, to have more VMs running on the hypervisor

For production setups, you might want to consider checking *Optimize performance in virtual environment* to get the best performance (except the section about CPU pinning).

The `crudini` package has to be installed.

**See also:**

For more information about:

- PCI passthrough, refer to <https://docs.openstack.org/nova/pike/admin/pci-passthrough.html>
- SR-IOV, refer to <https://docs.openstack.org/ocata/networking-guide/config-sriov.html>
- CPU pinning, refer to <https://docs.openstack.org/nova/pike/admin/cpu-topologies.html>
- Hugepages, refer to <https://docs.openstack.org/nova/pike/admin/huge-pages.html>

**Passthrough mode**

With this configuration, the Turbo Router VM will get dedicated interfaces.

The passthrough mode is only available if the compute node hardware supports Intel VT-d, and if it is enabled (see *enable Intel VT-d*).

1. **[Compute]** Get the vendor and product id of the dedicated interface that you want to give to the Turbo Router VM. In this example, for the eno1 interface, we have 8086 as vendor id and 1583 as product id. Please replace the interface name, pci id, vendor id and product id by your own values:

```
# IFACE=en01
# ethtool -i $IFACE | grep bus-info | awk '{print $2}'
0000:81:00.1
# PCI=0000:81:00.1
# lspci -n -s $PCI | awk '{print $3}'
8086:1583
# VENDOR_ID=8086
# PRODUCT_ID=1583
```

2. **[Compute]** Configure a PCI device alias. It will identify the vendor\_id and product\_id found in first step with the a1 alias in the next steps.

```
# crudini --set /etc/nova/nova.conf pci alias \
           '{ "vendor_id": "$VENDOR_ID", "product_id": "$PRODUCT_ID",
↵ "device_type": "type-PF", "name": "a1" }'
```

3. **[Compute]** Tell which PCI device can be given to VMs. Here we give the PCI device 0000:81:00.1:

```
# crudini --set /etc/nova/nova.conf pci passthrough_whitelist \
           '{ "address": "$PCI" }'
# service nova-compute restart
```

---

**Note:** It is possible to add more PCI devices here, by giving a list to crudini (i.e: ‘[{ “address”: “pci1” }, { “address”: “pci2” }]’) in the previous command.

---

4. **[Controller]** Export the previously configured variables, as well as the Turbo Router qcow2 file path:

```
# TURBO_QCOW2=/path/to/6wind-turbo-*-<arch>-<version>.qcow2
# IFACE=en01
# PCI=0000:81:00.1
# VENDOR_ID=8086
# PRODUCT_ID=1583
```

5. **[Controller]** Configure nova-scheduler to activate the PciPassthroughFilter. Note that if you have enabled filters already, you should just add PciPassthroughFilter to your list:

```
# crudini --set /etc/nova/nova.conf DEFAULT enabled_filters \
           'RetryFilter,AvailabilityZoneFilter,RamFilter,DiskFilter,
↪ComputeFilter,ComputeCapabilitiesFilter,ImagePropertiesFilter,
↪ServerGroupAntiAffinityFilter,ServerGroupAffinityFilter,PciPassthroughFilter'
# crudini --set /etc/nova/nova.conf DEFAULT available_filters \
           'nova.scheduler.filters.all_filters'
# service nova-scheduler restart
```

6. **[Controller]** Configure a PCI device alias. It will identify the vendor\_id and product\_id found in first step with the a1 alias:

```
# crudini --set /etc/nova/nova.conf pci alias \
           '{ "vendor_id":"'$VENDOR_ID'", "product_id":"'$PRODUCT_ID'",
↪"device_type":"type-PF", "name":"a1" }'
# service nova-api restart
```

7. **[Controller]** Use glance to create a VM image with the Turbo Router qcow2 file:

```
# openstack image create --disk-format qcow2 --container-format bare \
                          --file $TURBO_QCOW2 turbo-router
```

8. **[Controller]** Create a flavor with 8192MB of memory and 4 virtual CPUs.

```
# openstack flavor create --ram 8192 \
                          --vcpus 4 turbo-router-passthrough
```

9. **[Controller]** Configure the flavor to request 1 pci device in alias a1:

```
# openstack flavor set turbo-router-passthrough \
                          --property "pci_passthrough:alias="a1:1"
```

---

**Note:** To request X devices, change the previous command into “a1:X”.

---

10. **[Controller]** Configure the flavor to use one NUMA node and the same hyperthreads, and enable hugepages to get deterministic performances. OpenStack will choose CPUs and memory on the same NUMA node as

the NICs:

```
# openstack flavor set turbo-router-passthrough \
    --property hw:numa_nodes=1 \
    --property hw:cpu_policy=dedicated \
    --property hw:cpu_thread_policy=require \
    --property hw:mem_page_size=large
```

#### 11. [Controller] Boot the Turbo Router VM:

```
# openstack server create --flavor turbo-router-passthrough \
    --image turbo-router \
    turbo-router_vm
```

The next step is to perform your *first configuration*.

### SR-IOV mode

SR-IOV enables an Ethernet port to appear as multiple, separate, physical devices called Virtual Functions (VF). You will need compatible hardware, and Intel VT-d configured. The traffic coming from each VF can not be seen by the other VFs. The performance is almost as good as the performance in passthrough mode.

Being able to split an Ethernet port can increase the VM density on the hypervisor compared to passthrough mode.

In this configuration, the Turbo Router VM will get Virtual Functions (VFs).

#### See also:

For more information about SR-IOV, more advanced configurations, interconnecting physical and virtual networks, please check your OpenStack documentation: <https://docs.openstack.org/ocata/networking-guide/config-sriov.html>

1. [Compute] First check if the network interface that you want to use supports SR-IOV and how much VFs can be configured. Here we check for eno1 interface. Please export your own interface name instead of eno1.

```
# IFACE=eno1
# lspci -vvv -s $(ethtool -i $IFACE | grep bus-info | awk -F': ' '{print $2}') |
↳grep SR-IOV
    Capabilities: [160 v1] Single Root I/O Virtualization (SR-IOV)
# lspci -vvv -s $(ethtool -i $IFACE | grep bus-info | awk -F': ' '{print $2}') |
↳grep VFs
    Initial VFs: 64, Total VFs: 64, Number of VFs: 0, Function
↳Dependency Link: 00
```

2. [Compute] Add VFs, and check that those VFs were created. You should add this command to a custom startup script to make it persistent. Please export your own vf pci id instead of 81:0a.0.

```
# echo 2 > /sys/class/net/$IFACE/device/sriov_numvfs
# lspci | grep Ethernet | grep Virtual
81:0a.0 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev_
↪02)
81:0a.1 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev_
↪02)
# VF_PCI=0000:81:0a.0
```

3. **[Compute]** Get the vendor and product id of the dedicated VF that you want to give to the Turbo Router VM. In this example, for the 81:0a.0 VF, we have 8086 as vendor id and 154c as product id. Let's export the two variables `VENDOR_ID` and `PRODUCT_ID` for further use:

```
# lspci -n -s $VF_PCI | awk '{print $3}'
8086:154c
# VENDOR_ID=8086
# PRODUCT_ID=154c
```

4. **[Compute]** You need to set `eno1` up so that VFs are properly detected in the guest VM.

```
# ip link set $IFACE up
```

5. **[Compute]** Install and configure the SR-IOV agent:

```
# apt-get install neutron-sriov-agent
# crudini --set /etc/neutron/plugins/ml2/sriov_agent.ini securitygroup \
          firewall_driver neutron.agent.firewall.NoopFirewallDriver
# crudini --set /etc/neutron/plugins/ml2/sriov_agent.ini sriov_nic \
          physical_device_mappings physnet2:$IFACE
# service neutron-sriov-agent restart
```

6. **[Compute]** Configure a PCI device alias. It will identify the `vendor_id` and `product_id` found in first step with the `a1` alias. Also tell which PCI device can be given to VMs. Here we give all the VFs configured on `eno1`:

```
# crudini --set /etc/nova/nova.conf pci alias \
          '{ "vendor_id": "$VENDOR_ID", "product_id": "$PRODUCT_ID",
↪ "device_type": "type-VF", "name": "a1" }'
```

7. **[Compute]** Tell which PCI device can be given to VMs. Here we give all the VFs configured on `eno1`:

```
# crudini --set /etc/nova/nova.conf pci passthrough_whitelist \
          '{ "devname": "$IFACE", "physical_network": "physnet2" }'
# service nova-compute restart
```

8. **[Controller]** Export the previously configured variables, as well as the Turbo Router `qcow2` file path:

```
# TURBO_QCOW2=/path/to/6wind-turbo-*-<arch>-<version>.qcow2
# IFACE=en01
# VENDOR_ID=8086
# PRODUCT_ID=154c
```

9. **[Controller]** Configure nova-scheduler to activate the PciPassthroughFilter. Note that if you have enabled filters already, you should just add PciPassthroughFilter to your list:

```
# crudini --set /etc/nova/nova.conf DEFAULT enabled_filters \
    'RetryFilter,AvailabilityZoneFilter,RamFilter,DiskFilter,
↪ComputeFilter,ComputeCapabilitiesFilter,ImagePropertiesFilter,
↪ServerGroupAntiAffinityFilter,ServerGroupAffinityFilter,PciPassthroughFilter'
# crudini --set /etc/nova/nova.conf DEFAULT available_filters \
    'nova.scheduler.filters.all_filters'
# service nova-scheduler restart
```

10. **[Controller]** Configure a PCI device alias. It will identify the vendor\_id and product\_id found in first step with the a1 alias:

```
# crudini --set /etc/nova/nova.conf pci alias \
    '{ "vendor_id":"$VENDOR_ID", "product_id":"$PRODUCT_ID",
↪"device_type":"type-VF", "name":"a1" }'
# service nova-api restart
```

11. **[Controller]** Use glance to create a VM image with the Turbo Router qcow2 file:

```
# openstack image create --disk-format qcow2 --container-format bare \
    --file $TURBO_QCOW2 turbo-router
```

12. **[Controller]** Create a flavor with 8192MB of memory and 4 virtual CPUs.

```
# openstack flavor create --ram 8192 \
    --vcpus 4 turbo-router-sriov
```

13. **[Controller]** Configure the flavor to request 1 pci device in alias a1:

```
# openstack flavor set turbo-router-sriov \
    --property "pci_passthrough:alias"="a1:1"
```

14. **[Controller]** Configure the flavor to use one NUMA node and the same hyperthreads, and enable hugepages to get deterministic performances. OpenStack will choose CPUs and memory on the same NUMA node as the NICs:

```
# openstack flavor set turbo-router-sriov \
    --property hw:numa_nodes=1 \
    --property hw:cpu_policy=dedicated \
```

(continues on next page)

(continued from previous page)

```
--property hw:cpu_thread_policy=require \  
--property hw:mem_page_size=large
```

15. **[Controller]** Boot the Turbo Router VM:

```
# openstack server create --flavor turbo-router-sriov \  
--image turbo-router \  
turbo-router_vm
```

The next step is to perform your *first configuration*.

## 2.2.6 Install as a VM using VMware

### VMware basic deployment

Turbo Router is provided in the form of an OVA (Open Virtualization Appliance) file. It is supported on:

- ESX/ESXi 5.5 and later
- vCenter Server 5.5 and later
- Fusion 6.x
- Workstation 10.x
- Player 6.x

#### See also:

Refer to this [link](https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2007) (https://kb.vmware.com/selfservice/microsites/search.do?language=en\_US&cmd=displayKC&externalId=2007) and that [one](https://kb.vmware.com/selfservice/search.do?cmd=displayKC&docType=kc&docTypeID=DT_KB_1_1&externalId=2007) (https://kb.vmware.com/selfservice/search.do?cmd=displayKC&docType=kc&docTypeID=DT\_KB\_1\_1&externalId=2007) for compatibility. Turbo Router's hardware version is 10.

The image is configured to run with:

- 4 cores
- 8GB RAM
- 1 vmxnet3 NIC

If you wish to add other NICs, make sure they have the `vmxnet3 virtualDev` attribute, or Turbo Router will not be able to use them.

In order to boot your Turbo Router VM, import the OVA file in your VMware product.

The next step is to perform your *first configuration*.

**See also:**

Refer to VMware documentation for details on how to deploy VM images. For instance *Deploying using vSphere 6.5, ESXi 6.5 or vCenter Server 6.5* ([https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.vm\\_admin.doc/GUID-AFEDC48B-C96F-4088-9C1F-4F0A30E965DE.html](https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.vm_admin.doc/GUID-AFEDC48B-C96F-4088-9C1F-4F0A30E965DE.html))

**VMware performance tuning****All ESXi version**

Optimizations must be done in the hypervisor to achieve the best performance.

In the **Virtual Hardware** tab of the VM settings, set:

- VM CPU Reservation field to its maximal value
- VM CPU Limit field to **Unlimited**

In the **VM Options** tab, **Advanced** part of the VM settings, set:

- `sched.cpu.latencySensitivity` to 'High': used to ensure pinning and exclusive affinity of all CPUs of a VNF

**ESXi 6.5 and newer versions**

Since ESXi 6.5, new tuning options are available to improve hypervisor's performance. Before going further, all the settings described in the previous section must be applied.

In the **VM Options** tab, **Advanced** part of the VM settings, press the **Configuration Parameters** button to set:

- `ethernetX.ctxPerDev` to 1 (where `ethernetX` is the NIC which will be handled by the Turbo Router): each NIC configured with `ctxPerDev` will receive a TX thread in the hypervisor. It can be checked in the `esxtop` output. The `ctxPerDev` recommendation must be enabled for NICs that are expected to process an high packet load.
- `sched.cpu.latencySensitivity.sysContexts` to numerical value: system threads (TX and RX) are assigned exclusive physical CPU cores. The numerical value assigned to `sched.cpu.latencySensitivity.sysContexts` must equal the number of active threads for the VNF. For example, if one receive thread exists and three TX threads have been set using the `ctxPerDev` command, the value set must be 4. In this example, 4 physical CPU cores must be available and unreserved.

More details are available in VMware document regarding high performance setups (<https://www.vmware.com/techpapers/2017/tuning-vmware-vcloud-nfv-for-data-plane-intensive-workloads.html>).

### esxtop reading

First, run esxtop command in the hypervisor’s console.

Here is the default esxtop screen (also accessible by hitting ‘c’):

```

4:53:33pm up 12 days 8:06, 654 worlds, 2 VMs, 5 vCPUs; CPU load average: 0.24, 0.05,
↪0.02
PCPU USED(%): 0.0 0.4 0.0 0.2 2.9 0.1 0.1 1.6 0.1 0.0 118 0.0 0.0 0.0 0.1 0.0 0.0 0.2
↪112 0.0 0.1 1.7 0.0 0.2 AVG: 9.9
PCPU UTIL(%): 0.1 100 0.1 0.3 2.5 0.1 0.2 1.5 0.1 0.1 100 0.1 0.1 0.1 0.1 0.1 0.1 0.2
↪100 0.1 0.2 1.6 0.1 0.3 AVG: 12
CORE UTIL(%): 100 0.3 2.6 1.6 0.2 100 0.2 0.2 0.3
↪100 1.7 0.2 AVG: 25

      ID      GID NAME                                NWLD  %USED  %RUN  %SYS
↪%WAIT %VMWAIT  %RDY  %IDLE  %OVRP  %CSTP  %MLMTD  %SWPWT
685528 685528 6WIND-TI                                11 237.16 301.35 0.00
↪803.45 0.00 0.01 0.00 0.02 0.00 0.00 0.00
 21609 21609 VMware vCenter Server Appliance    13 3.59 3.08 0.02
↪1300.00 0.00 0.02 198.30 0.01 0.00 0.00 0.00
685520 685520 esxtop.228984                          1 2.87 2.46 0.00
↪97.97 - 0.00 0.00 0.00 0.00 0.00 0.00
 1      1 system                                270 0.42 2103.44 0.00
↪24709.04 - 307.76 0.00 0.28 0.00 0.00 40.78
10304 10304 vpxa.67910                                24 0.17 0.15 0.00
↪2400.00 - 0.00 0.00 0.00 0.00 0.00 0.00
 5662 5662 hostd.67290                            24 0.12 0.09 0.04
↪2400.00 - 0.00 0.00 0.02 0.00 0.00 0.00
 8      8 helper                                142 0.02 0.03 0.00
↪14200.00 - 0.01 0.00 0.00 0.00 0.00 0.00
 4241 4241 ioFilterVPServer.67102                2 0.02 0.02 0.00
↪200.00 - 0.00 0.00 0.00 0.00 0.00 0.00
685432 685432 sshd.228973                            1 0.02 0.02 0.00
↪100.00 - 0.00 0.00 0.00 0.00 0.00 0.00
 10     10 ft                                  4 0.01 0.01 0.00
↪400.00 - 0.00 0.00 0.00 0.00 0.00 0.00
    
```

Threads (including ctxPerDev) threads can be displayed by hitting ‘e’, with the GID number of the process. You can check here the number of threads created for the VM, and their current load:

```

4:55:29pm up 12 days 8:08, 654 worlds, 2 VMs, 5 vCPUs; CPU load average: 0.26, 0.15,
↪0.05
PCPU USED(%): 0.0 0.4 0.0 0.0 2.3 0.0 0.1 0.2 0.2 0.0 113 0.0 0.0 2.2 0.0 0.0 0.0 2.7
↪118 0.0 0.0 0.0 0.0 0.1 AVG: 10
PCPU UTIL(%): 0.1 100 0.1 0.1 2.2 0.1 0.1 0.3 0.2 0.1 100 0.1 0.1 2.0 0.1 0.1 0.1 2.4
↪100 0.1 0.1 0.1 0.1 0.1 AVG: 12
    
```

(continues on next page)

(continued from previous page)

```

CORE UTIL(%): 100    0.3    2.3    0.4    0.4    100    2.1    0.1    2.5
↪100    0.3    0.3    AVG: 25

      ID      GID NAME      NWLD  %USED  %RUN  %SYS
↪%WAIT %VMWAIT  %RDY  %IDLE %OVRLP %CSTP %MLMTD %SWPWT
228985  685528 vmx          1    0.01   0.00  0.00
↪100.00    -    0.00  0.00  0.00  0.00  0.00  0.00
228987  685528 NetWorld-VM-228986  1    0.00   0.00  0.00
↪100.00    -    0.00  0.00  0.00  0.00  0.00  0.00
228988  685528 vmast.228986  1    0.00   0.00  0.00
↪100.00    -    0.00  0.00  0.00  0.00  0.00  0.00
228991  685528 vmx-vthread-7  1    0.00   0.00  0.00
↪100.00    -    0.00  0.00  0.00  0.00  0.00  0.00
228993  685528 vmx-mks:6WIND-TI  1    0.01   0.01  0.00
↪100.00    -    0.00  0.00  0.00  0.00  0.00  0.00
228994  685528 vmx-svga:6WIND-TI  1    0.02   0.02  0.00
↪100.00    -    0.01  0.00  0.00  0.00  0.00  0.00
228998  685528 vmx-vcpu-0:6WIND-TI  1    0.41  100.17  0.00
↪0.00    0.00  0.00  0.00  0.01  0.00  0.00  0.00
228999  685528 vmx-vcpu-1:6WIND-TI  1  113.65  100.17  0.00
↪0.00    0.00  0.00  0.00  0.00  0.00  0.00  0.00
229000  685528 vmx-vcpu-2:6WIND-TI  1  118.87  100.17  0.00
↪0.00    0.00  0.00  0.00  0.00  0.00  0.00  0.00
229170  685528 NetWorld-Dev-67108888-Tx  1    0.00   0.00  0.00
↪100.00    -    0.00  0.00  0.00  0.00  0.00  0.00
229171  685528 NetWorld-Dev-50331672-Tx  1    0.00   0.00  0.00
↪100.00    -    0.00  0.00  0.00  0.00  0.00  0.00
21609  21609 VMware vCenter Server Appliance  13    4.66   4.01  0.02
↪1298.06  0.00  0.08  196.53  0.01  0.00  0.00  0.00
    
```

The network screen (accessible by hitting ‘n’) is really useful to check if the hypervisor is dropping packets:

```

5:00:32pm up 12 days 8:13, 649 worlds, 2 VMs, 5 vCPUs; CPU load average: 0.26, 0.26,
↪0.14

      PORT-ID USED-BY      TEAM-PNIC DNAME      PKTTX/s  MbTX/s
↪ PSZTX  PKTRX/s  MbrX/s  PSZRX %DRPTX %DRPRX
33554433 Management      n/a vSwitch0      0.00    0.00
↪ 0.00    0.00    0.00    0.00  0.00  0.00
33554434 vmnic0          - vSwitch0      6.65    0.01
↪ 229.00  6.46    0.01  145.00  0.00  0.00
33554435 Shadow of vmnic0  n/a vSwitch0      0.00    0.00
↪ 0.00    0.00    0.00    0.00  0.00  0.00
33554436 vmk0          vmnic0 vSwitch0      6.65    0.02
↪ 335.00  6.06    0.01  131.00  0.00  0.00
    
```

(continues on next page)

(continued from previous page)

|          |                                |                    |        |      |      |
|----------|--------------------------------|--------------------|--------|------|------|
| 33554438 | 69973:VMware vCenter Server Ap | vmnic0 vSwitch0    | 4.70   | 0.01 |      |
| ↪ 189.00 | 4.89                           | 0.01               | 355.00 | 0.00 | 0.00 |
| 33554463 | 228986:6WIND-VA-1.6.2-1        | vmnic0 vSwitch0    | 0.00   | 0.00 |      |
| ↪ 0.00   | 1.96                           | 0.00               | 117.00 | 0.00 | 0.00 |
| 50331649 | Management                     | n/a DvsPortset-0   | 0.00   | 0.00 |      |
| ↪ 0.00   | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 50331650 | LACP_MgmtPort                  | n/a DvsPortset-0   | 0.00   | 0.00 |      |
| ↪ 0.00   | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 50331651 | lag1                           | n/a DvsPortset-0   | 0.00   | 0.00 |      |
| ↪ 0.00   | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 50331652 | vmnic7                         | - DvsPortset-0     | 0.00   | 0.00 |      |
| ↪ 0.00   | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 50331653 | Shadow of vmnic7               | n/a DvsPortset-0   | 0.00   | 0.00 |      |
| ↪ 0.00   | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 50331654 | vmnic6                         | - DvsPortset-0     | 0.20   | 0.00 |      |
| ↪ 124.00 | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 50331655 | Shadow of vmnic6               | n/a DvsPortset-0   | 0.00   | 0.00 |      |
| ↪ 0.00   | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 50331656 | vmnic5                         | - DvsPortset-0     | 0.20   | 0.00 |      |
| ↪ 124.00 | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 50331657 | Shadow of vmnic5               | n/a DvsPortset-0   | 0.00   | 0.00 |      |
| ↪ 0.00   | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 50331658 | vmnic4                         | - DvsPortset-0     | 0.20   | 0.00 |      |
| ↪ 124.00 | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 50331659 | Shadow of vmnic4               | n/a DvsPortset-0   | 0.00   | 0.00 |      |
| ↪ 0.00   | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 50331672 | 228986:6WIND-TI.eth2           | lag1* DvsPortset-0 | 0.00   | 0.00 |      |
| ↪ 0.00   | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 67108865 | Management                     | n/a DvsPortset-1   | 0.00   | 0.00 |      |
| ↪ 0.00   | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 67108888 | 228986:6WIND-TI.eth1           | void DvsPortset-1  | 0.00   | 0.00 |      |
| ↪ 0.00   | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 83886081 | Management                     | n/a DvsPortset-2   | 0.00   | 0.00 |      |
| ↪ 0.00   | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |
| 83886087 | 228986:6WIND-TI.eth3           | void DvsPortset-2  | 0.00   | 0.00 |      |
| ↪ 0.00   | 0.00                           | 0.00               | 0.00   | 0.00 | 0.00 |

The column details can be checked in the esxtop statistics reading guide (<https://communities.vmware.com/docs/DOC-9279>).

## 2.2.7 Install as a VM using Proxmox VE

This chapter explains how to start a Turbo Router VM using Proxmox VE and the `.iso` file.

It expects that you already installed a Proxmox VE cluster, in which you are able to spawn VMs with network connected.

It follows the following steps:

- make the `.iso` file available to Proxmox VE
- create and configure a VM
- boot the VM using the `.iso` file
- install Turbo Router on the virtual disk

### Upload the `.iso` file

Select the local storage of your node in the left pane and visualize its content:

The screenshot shows the Proxmox VE web interface. The top navigation bar includes the Proxmox logo, version 'Virtual Environment 5.2-6', a search bar, and user information 'You are logged in as 'root@pam''. Action buttons for 'Documentation', 'Create VM', 'Create CT', and 'Logout' are visible. The left sidebar shows a tree view with 'Datacenter' expanded to 'berlin', and 'local (berlin)' selected. The main content area is titled 'Storage 'local' on node 'berlin'' and contains tabs for 'Summary', 'Content', and 'Permissions'. The 'Content' tab is active, showing a table with columns 'Name', 'Format', 'Type', and 'Size', which is currently empty. Below the main content area, there are two tabs: 'Tasks' and 'Cluster log'. The 'Cluster log' tab is active, displaying a table with columns 'Start Time', 'End Time', 'Node', 'User name', 'Description', and 'Status'. The table contains five entries, all with a status of 'OK'.

| Start Time ↓    | End Time        | Node   | User name | Description | Status |
|-----------------|-----------------|--------|-----------|-------------|--------|
| Sep 05 17:36:10 | Sep 05 17:36:22 | berlin | root@pam  | Shell       | OK     |
| Sep 05 17:36:08 | Sep 05 17:36:28 | berlin | root@pam  | Shell       | OK     |
| Sep 05 17:36:06 | Sep 05 17:36:27 | berlin | root@pam  | Shell       | OK     |
| Sep 05 17:36:03 | Sep 05 17:36:14 | berlin | root@pam  | Shell       | OK     |
| Sep 05 17:35:23 | Sep 05 17:35:41 | berlin | root@pam  | Shell       | OK     |

Press the Upload button. In the pop-up window, select ISO image as content type and point to the Turbo Router .iso file on your local disk. Then press Upload to send this file to your Proxmox VE node:

The screenshot shows the Proxmox VE interface. At the top, it says 'PROXMOX Virtual Environment 5.2-6'. The user is logged in as 'root@pam'. The main area is titled 'Storage 'local' on node 'berlin''. There are tabs for 'Summary', 'Content', and 'Permissions'. The 'Content' tab is active, showing a table with columns 'Name', 'Format', 'Type', and 'Size'. An 'Upload' dialog box is open in the center, with 'Content' set to 'ISO image' and a file path 'C:\fakepath\@wind-turbo-router-'. Below the dialog is a 'Tasks' table with 5 rows of shell command logs.

| Start Time ↓    | End Time        | Node   | User name | Description | Status |
|-----------------|-----------------|--------|-----------|-------------|--------|
| Sep 05 17:36:10 | Sep 05 17:36:22 | berlin | root@pam  | Shell       | OK     |
| Sep 05 17:36:08 | Sep 05 17:36:28 | berlin | root@pam  | Shell       | OK     |
| Sep 05 17:36:06 | Sep 05 17:36:27 | berlin | root@pam  | Shell       | OK     |
| Sep 05 17:36:03 | Sep 05 17:36:14 | berlin | root@pam  | Shell       | OK     |
| Sep 05 17:35:23 | Sep 05 17:35:41 | berlin | root@pam  | Shell       | OK     |

The .iso file is now available to this node:

The screenshot shows the Proxmox VE 5.2-6 interface. The top navigation bar includes the Proxmox logo, version information, a search bar, and user login details ('root@pam'). There are buttons for 'Documentation', 'Create VM', 'Create CT', and 'Logout'. The main content area is titled 'Storage 'local' on node 'berlin'' and features a sidebar with a tree view showing 'Datacenter' > 'berlin' > 'local (berlin)'. The main panel has tabs for 'Summary', 'Content', and 'Permissions'. The 'Content' tab is active, displaying a table of ISO images:

| Name                                | Format | Type      | Size       |
|-------------------------------------|--------|-----------|------------|
| ISO image (1 Item)                  |        |           |            |
| 6wind-turbo-router-x86_64-2.0.0.iso | iso    | ISO image | 309.09 MIB |

Below the storage view, there is a 'Tasks' section with a 'Cluster log' table:

| Start Time ↓    | End Time        | Node   | User name | Description | Status |
|-----------------|-----------------|--------|-----------|-------------|--------|
| Sep 05 17:38:32 | Sep 05 17:38:33 | berlin | root@pam  | Copy data   | OK     |
| Sep 05 17:36:10 | Sep 05 17:36:22 | berlin | root@pam  | Shell       | OK     |
| Sep 05 17:36:08 | Sep 05 17:36:28 | berlin | root@pam  | Shell       | OK     |
| Sep 05 17:36:06 | Sep 05 17:36:27 | berlin | root@pam  | Shell       | OK     |
| Sep 05 17:36:03 | Sep 05 17:36:14 | berlin | root@pam  | Shell       | OK     |

## Create and boot the VM

In the top right corner, press the **Create VM** button to launch the creation wizard. In **General** tab, check the node and the VM ID, and give a name to the VM, then press **Next**:

### Create: Virtual Machine ⓧ

- General**
- OS
- Hard Disk
- CPU
- Memory
- Network
- Confirm

Node:  Resource Pool:

VM ID:

Name:

? Help Advanced  Back Next

In OS tab, make sure to use the uploaded .iso file as CD/DVD and to specify a Linux with 4.X/3.X/2.X kernel as Guest OS, then press Next:

**Create: Virtual Machine**

General **OS** Hard Disk CPU Memory Network Confirm

Use CD/DVD disc image file (iso)  
Storage: local  
ISO image: 6wind-turbo-router-x86\_64-2.0

Guest OS:  
Type: Linux  
Version: 4.X/3.X/2.6 Kernel

Use physical CD/DVD Drive  
 Do not use any media

Advanced  **Back** **Next**

In Hard Disk tab, keep the default qcow2 device with VirtIO SCSI storage and allocate at least 10GB, then press Next:

The screenshot shows a 'Create: Virtual Machine' dialog box with the 'Hard Disk' tab selected. The dialog has a title bar with a close button and a tabbed interface with the following tabs: General, OS, Hard Disk (selected), CPU, Memory, Network, and Confirm. The 'Hard Disk' tab contains the following fields:

- Bus/Device:  (dropdown)
- Cache:  (dropdown)
- SCSI Controller: **VirtIO SCSI**
- Storage:  (dropdown)
- Disk size (GiB):  (spinner)
- Format:  (dropdown)

At the bottom of the dialog, there is a 'Help' button with a question mark icon, an 'Advanced' checkbox which is currently unchecked, and 'Back' and 'Next' buttons.

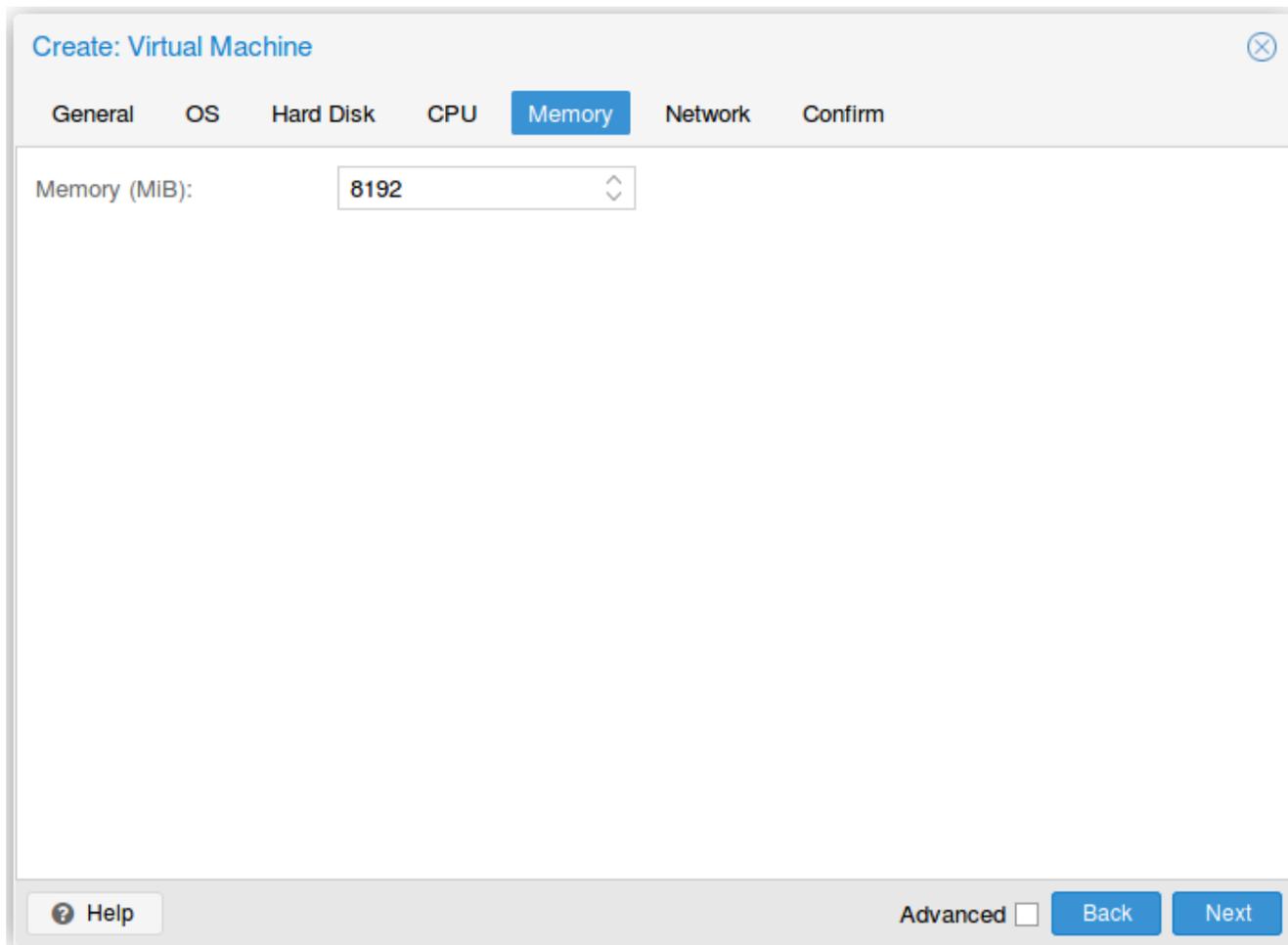
In CPU tab, allocate at least 2 cores and select host as CPU type, then press Next:

The screenshot shows a 'Create: Virtual Machine' wizard window with the 'CPU' tab selected. The window has a title bar with a close button. Below the title bar is a navigation bar with tabs: 'General', 'OS', 'Hard Disk', 'CPU' (highlighted), 'Memory', 'Network', and 'Confirm'. The main area contains the following fields:

|          |                                |              |                                   |
|----------|--------------------------------|--------------|-----------------------------------|
| Sockets: | <input type="text" value="1"/> | Type:        | <input type="text" value="host"/> |
| Cores:   | <input type="text" value="2"/> | Total cores: | 2                                 |

At the bottom of the window, there is a 'Help' button with a question mark icon, an 'Advanced' checkbox which is unchecked, and 'Back' and 'Next' buttons.

In Memory tab, allocate at least 8GB of RAM, then press Next:



In **Network** tab, bind the virtual management interface to a host bridge in order to have access to external network. Select **VirtIO** as model type, then press **Next**:

### Create: Virtual Machine

General OS Hard Disk CPU Memory **Network** Confirm

No network device

Bridge:  Model:

VLAN Tag:  MAC address:

Firewall:

Advanced

In **Confirm** tab, review your settings and press **Finish** to finalize the creation and get back to the main dashboard:

Create: Virtual Machine ✕

General OS Hard Disk CPU Memory Network **Confirm**

| Key ↑    | Value   |
|----------|---|
| cores    | 2   |
| cpu      | host  |
| ide2     | local:iso/6wind-turbo-router-x86_64-2.0.0.iso,media=cdrom |
| memory   | 8192  |
| name     | 6WIND-Turbo-Router  |
| net0     | virtio,bridge=vibr0                                       |
| nodename | berlin  |
| numa     | 0   |
| ostype   | l26   |
| scsi0    | local:32,format=qcow2                                     |
| scsihw   | virtio-scsi-pci   |
| sockets  | 1   |
| vmid     | 100   |

Start after created

Advanced  **Back** **Finish**

The VM is now available in the left pane below your physical node. Select it and review its hardware configuration:

Virtual Environment 5.2-6 Search You are logged in as 'root@pam' Documentation Create VM Create CT Logout

Server View Virtual Machine 100 (6WIND-Turbo-Router) on node 'berlin' Start Shutdown Console More Help

Summary Add Remove Edit Resize disk Move disk Revert

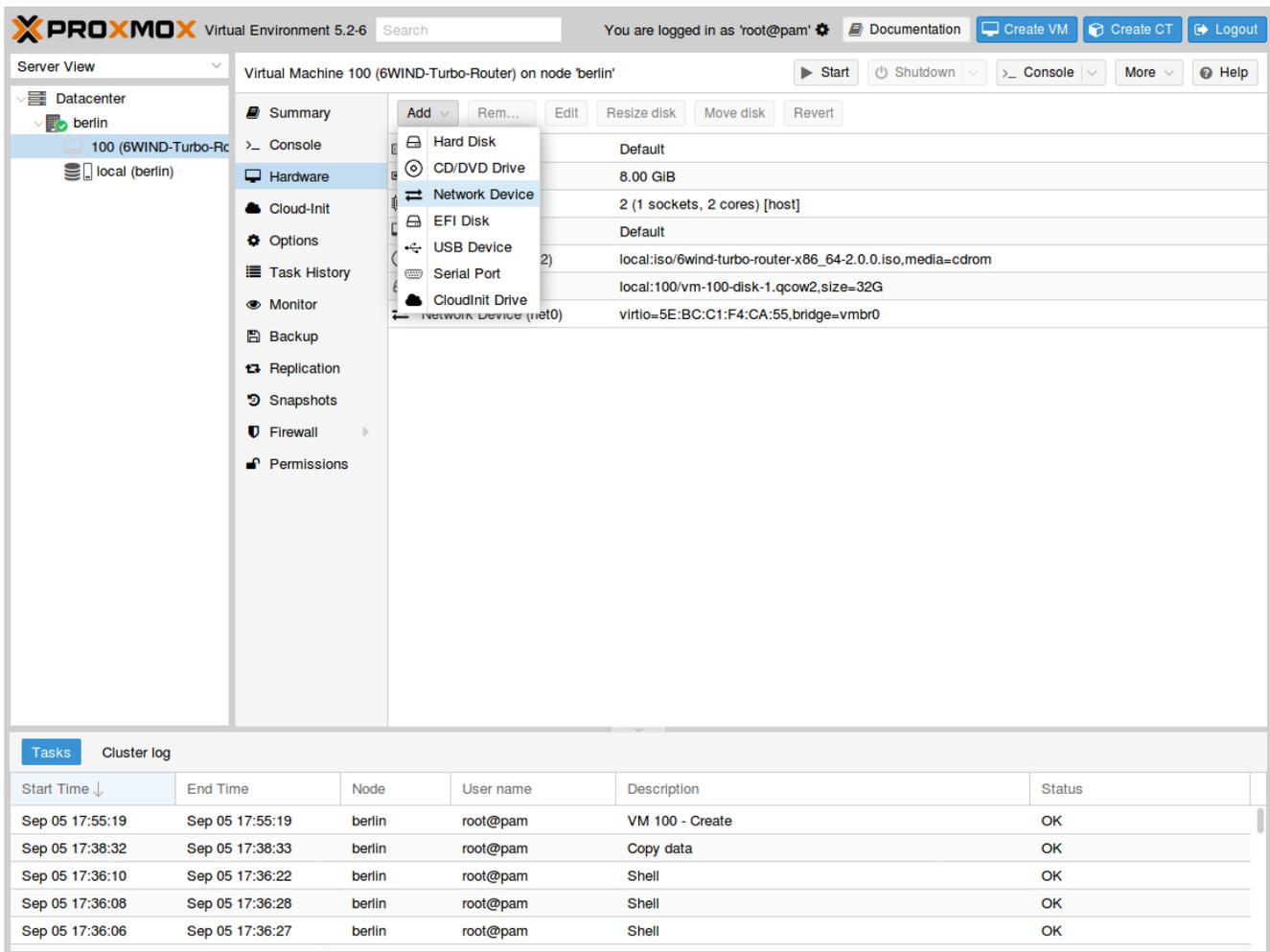
- Console
- Hardware
- Cloud-Init
- Options
- Task History
- Monitor
- Backup
- Replication
- Snapshots
- Firewall
- Permissions

|                       |  |
|-----------------------|--|
| Keyboard Layout       | Default  |
| Memory                | 8.00 GiB   |
| Processors            | 2 (1 sockets, 2 cores) [host]                              |
| Display               | Default  |
| CD/DVD Drive (ide2)   | local:iso/6wind-turbo-router-x86_64-2.0.0.iso,media=cdrrom |
| Hard Disk (scsi0)     | local:100/vm-100-disk-1.qcow2,size=32G                     |
| Network Device (net0) | virtio=5E:BC:C1:F4:CA:55,bridge=vibr0                      |

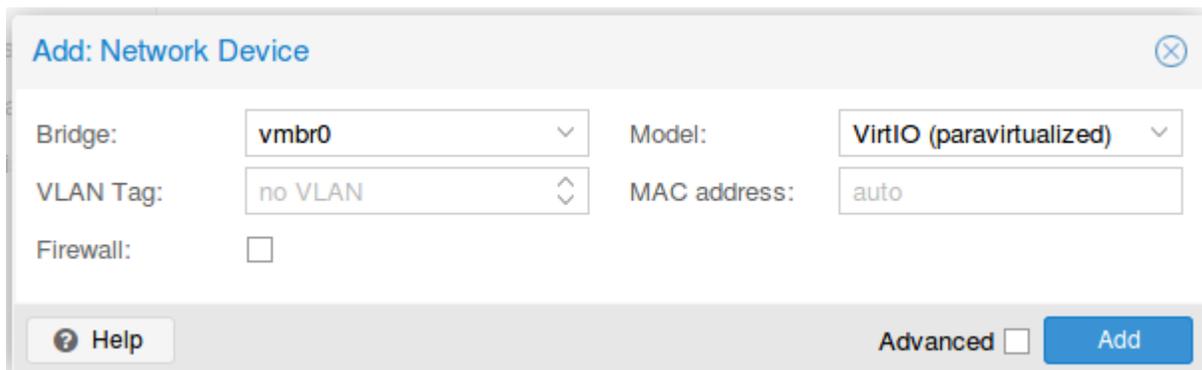
Tasks Cluster log

| Start Time ↓    | End Time        | Node   | User name | Description     | Status |
|-----------------|-----------------|--------|-----------|-----------------|--------|
| Sep 05 17:55:19 | Sep 05 17:55:19 | berlin | root@pam  | VM 100 - Create | OK     |
| Sep 05 17:38:32 | Sep 05 17:38:33 | berlin | root@pam  | Copy data       | OK     |
| Sep 05 17:36:10 | Sep 05 17:36:22 | berlin | root@pam  | Shell           | OK     |
| Sep 05 17:36:08 | Sep 05 17:36:28 | berlin | root@pam  | Shell           | OK     |
| Sep 05 17:36:06 | Sep 05 17:36:27 | berlin | root@pam  | Shell           | OK     |

Press Add > Network Device:



In the pop-up window, select an attachment bridge and choose VirtIO as model, then press Add:



The second network device can now be seen in the hardware configuration of the VM:

| Start Time ↓    | End Time        | Node   | User name | Description     | Status |
|-----------------|-----------------|--------|-----------|-----------------|--------|
| Sep 05 17:55:19 | Sep 05 17:55:19 | berlin | root@pam  | VM 100 - Create | OK     |
| Sep 05 17:38:32 | Sep 05 17:38:33 | berlin | root@pam  | Copy data       | OK     |
| Sep 05 17:36:10 | Sep 05 17:36:22 | berlin | root@pam  | Shell           | OK     |
| Sep 05 17:36:08 | Sep 05 17:36:28 | berlin | root@pam  | Shell           | OK     |
| Sep 05 17:36:06 | Sep 05 17:36:27 | berlin | root@pam  | Shell           | OK     |

**Warning:** Please make sure that there is no other Turbo Router live CDROM or live USB inserted in this VM. Otherwise the system might fail to boot properly.

Press Start in the top right corner to actually start the VM.

The next step consists in *installing on the virtual disk*.

## Install Turbo Router

**Warning:** Please carefully check the device associated to the disk you want to use, or you could wipe the wrong drive in the next step. When following this installation guide you have only one disk attached to the VM. Thus the device name is `sda`. If you attach additional virtual disks, make sure to choose the right device.

---

**Note:** Please make sure to select this disk as boot device after installation. You can access boot menu by pressing ESC at startup in the VM console.

---

Once the VM has booted on the `.iso` file, select it in the left pane of the main dashboard and press the `>_ Console` button to get access to the serial console.

Log in as `admin`, password `admin`, and at the prompt, do:

```
vrouter> cmd system-image install-on-disk sda
```

This command will install Turbo Router on `/dev/sda`. The relevant configuration files will be copied to the local drive.

---

**Note:** To restore from a backup file, add `backup-url <url>` to the previous command. This will restore your configurations, private keys, certificates and licenses.

The backup file must have been generated on the same or previous minor version (e.g. a backup from 3.0.1 can be restored on 3.0.x or 3.1.x).

---

Reboot to finally boot Turbo Router from the virtual hard disk:

```
vrouter> cmd reboot
```

The next step is to perform your *first configuration*.

### 2.2.8 Install as a VM using AWS

The Turbo Router private AMI image provides a simple way to deploy Turbo Router in AWS. Access to the AMI image must be requested to the 6WIND support team through the customer zone.

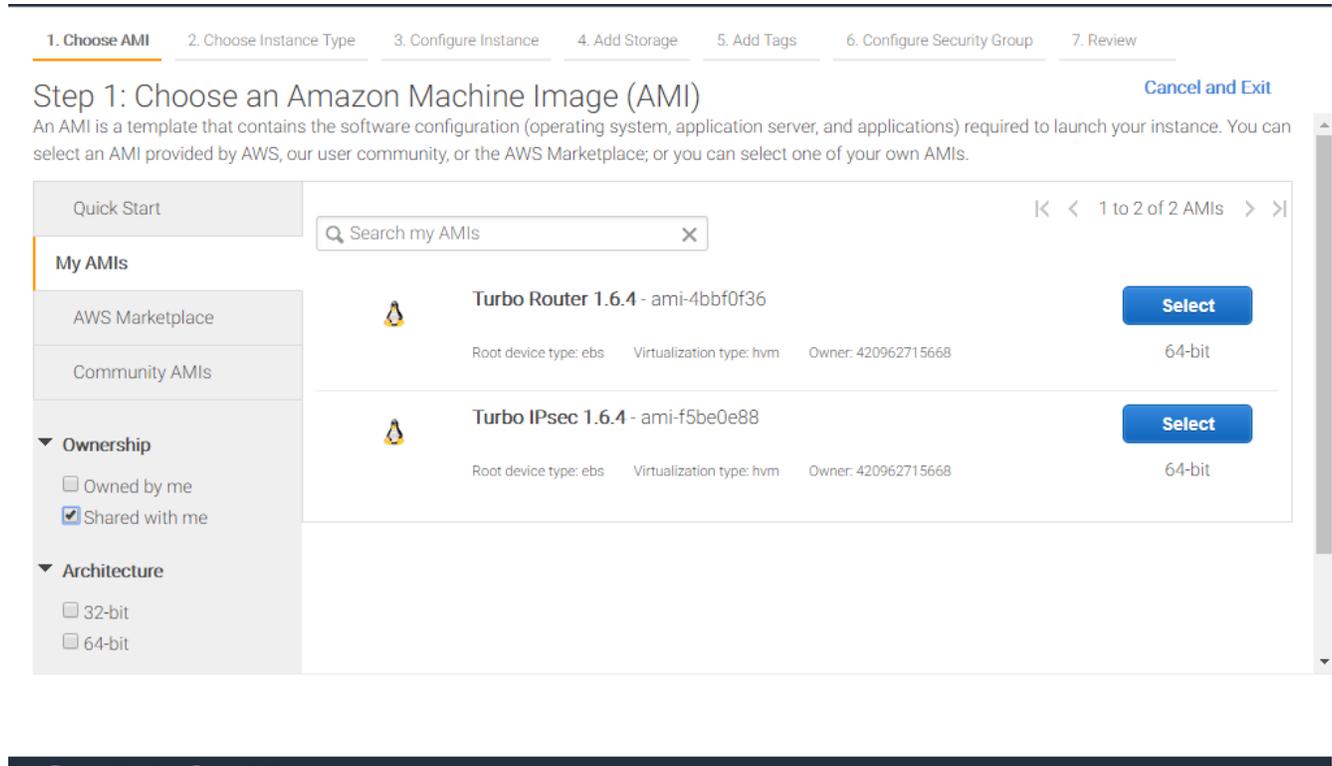
Once access is granted, the Turbo Router AMI will be available in the AWS management console when selecting `AMIs > Private Images`.

## Launch AWS Instance

From the EC2 homepage, select **Instances > Launch Instance**.

### Step 1: choose AMI

Select the Turbo AMI in **My AMIs > Ownership > Shared with me**.



## Step 2: choose instance type

This AMI requires either Intel 82599 VF adapters or ENA adapters. Please make sure to select an instance type (<https://aws.amazon.com/premiumsupport/knowledge-center/enable-configure-enhanced-networking/>) that supports these adapters.

## Step 3: configure instance

In AWS, console access is provided through the network and relies on cloud-init. cloud-init configuration must be provided in **Advanced Details > User data**.

### Step 3: Configure Instance Details

IAM role ⓘ None [Create new IAM role](#)

Shutdown behavior ⓘ Stop

Enable termination protection ⓘ  Protect against accidental termination

Monitoring ⓘ  Enable CloudWatch detailed monitoring  
Additional charges apply.

Tenancy ⓘ Shared - Run a shared hardware instance  
Additional charges will apply for dedicated tenancy.

T2/T3 Unlimited ⓘ  Enable  
Additional charges may apply

▼ Advanced Details

User data ⓘ  As text  As file  Input is already base64 encoded

No file selected.

In the following example, we pre-install the license file (make sure you replace the contents by your own). We also upload a startup configuration for the CLI.

This sample CLI configuration fulfills the minimal requirements to start Turbo Router with high performance. It consists in enabling DHCP on the first network interface, dedicating that interface to the FAST PATH (The fast path is the |turbo| component in charge of packet processing.) and enabling VLAN stripping.

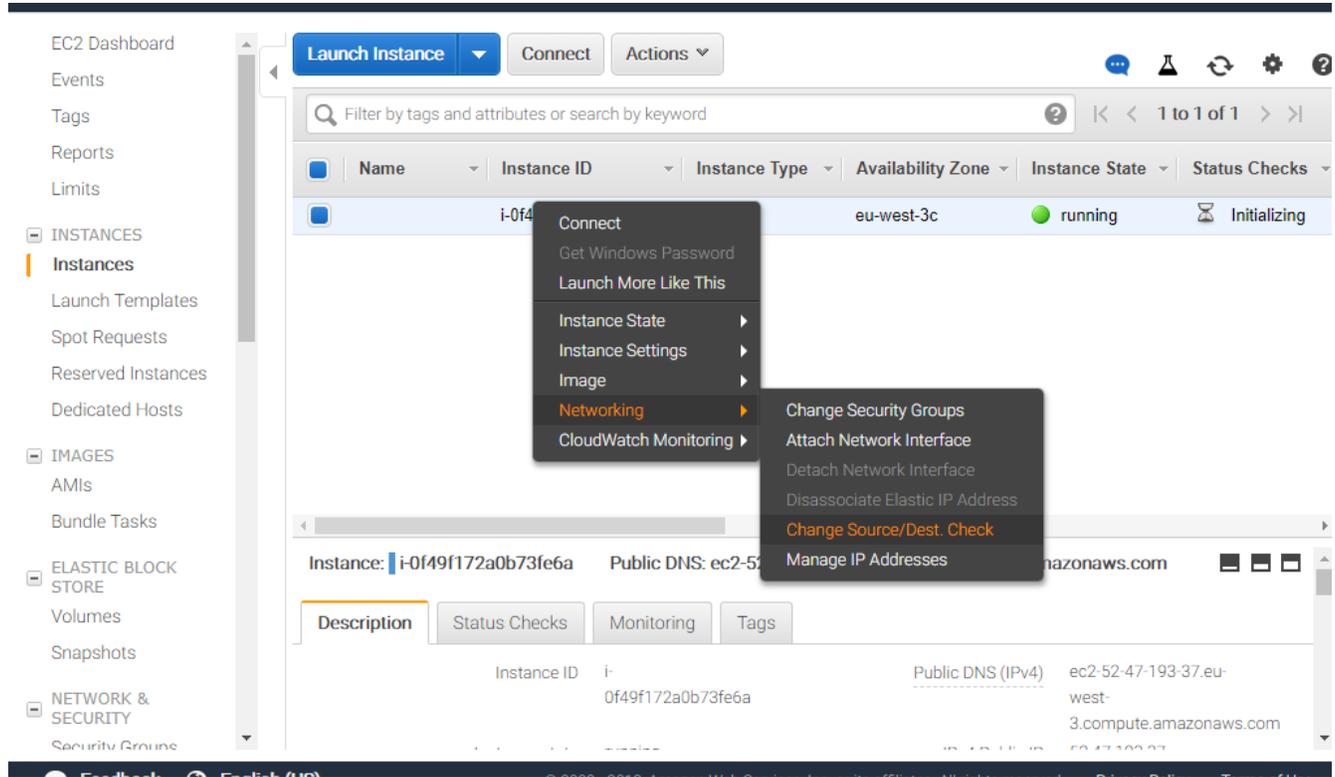
```
#cloud-config
write_files:
- path: /run/vrouter.startup
  content: |
    {
      "vrouter:config": {
        "vrouter-system:system": {
          "vrouter-license:license": {
            "online": {
              "serial": "xxx"
            }
          }
        }
      }
    }
```

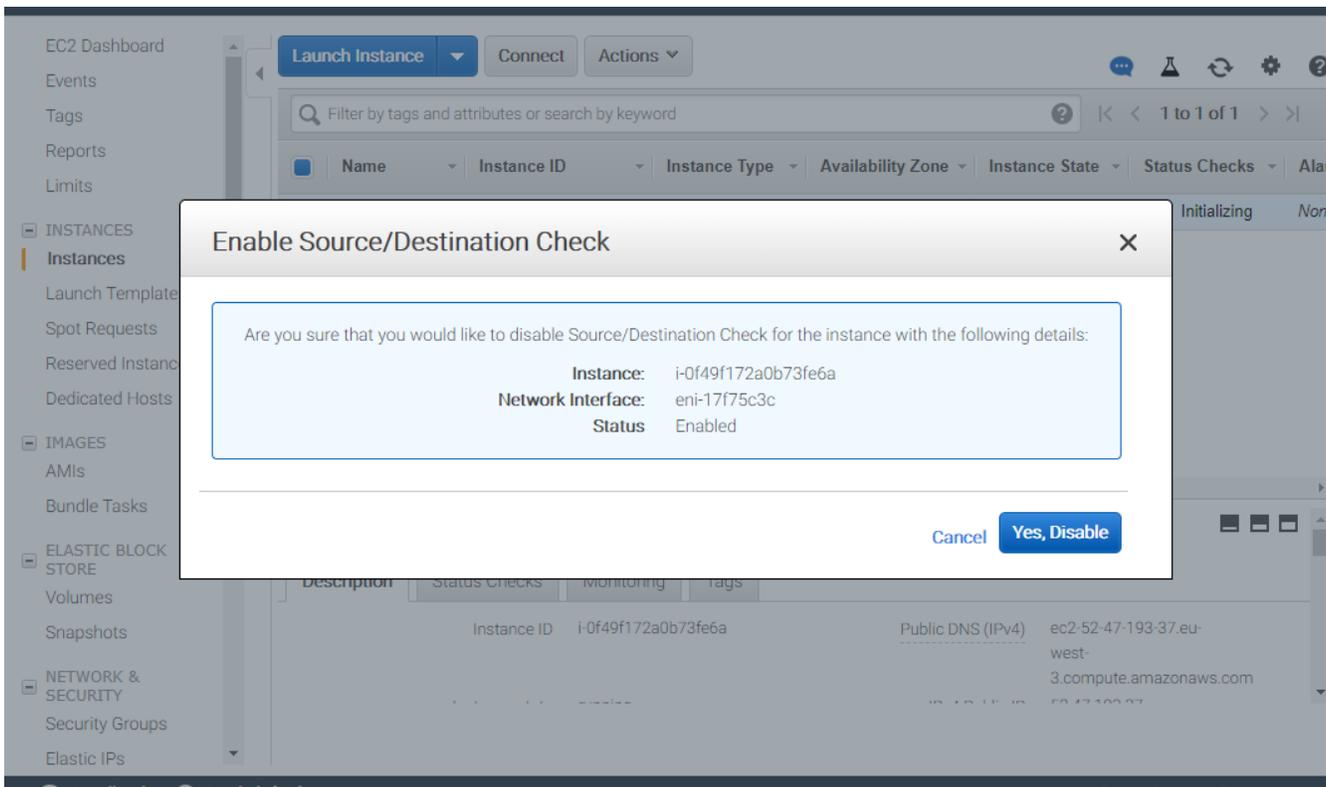
(continues on next page)



### Activate AWS IP forwarding

By default, AWS forbids IP forwarding. It must be enabled from the management console after the instance is launched as follows.





The next step is to perform your *first configuration*.

### 2.2.9 Update an existing installation

This section covers the software update of a Turbo Router appliance.

#### Prerequisites

---

**Note:** In Turbo Router version X.Y.Z, digits are referred to as follows:

- X = major
- Y = minor
- Z = maintenance

- 
- Update between maintenance versions within the same minor version is supported.
  - Update between consecutive minor versions is supported.
  - Update between major versions requires a fresh install.

- Update between inconsecutive minor versions requires to update in several steps between consecutive minor versions.
- Downgrade is not supported.

Examples:

| Update from | To    | Supported   |
|-------------|-------|---|
| 3.0.1       | 3.0.5 | <b>Yes</b>  |
| 3.0.5       | 3.1.0 | <b>Yes</b>  |
| 2.2.6       | 3.0.0 | No; backup your configurations, install from scratch and restore your configurations      |
| 3.0.5       | 3.2.0 | No; update to the latest 3.1.x, then from 3.1.x to 3.2.0.                                 |
| 3.0.0       | 2.2.6 | No; install from scratch. Configuration backup/restore is not supported when downgrading. |

The rationale for these restrictions is to properly support *API deprecation*.

## Backup existing files

If you are updating from Turbo Router 2.x, make sure that you update to the latest 2.x version first. Refer to Turbo Router 2.x documentation.

1. Before updating to a new revision, ensure that you are not using a *deprecated API*, at least in the *startup* configuration:

```
vrouter> validate startup
OK.
```

---

**Note:** Any use of deprecated or obsolete API should be fixed. It is advised to also check the saved configurations.

---

2. Backup your existing configurations, private keys, certificates and licenses.

```
vrouter> cmd backup export url <backup-url>
OK.
```

**Warning:** The backup file contains sensitive data, like private keys. Make sure to keep it private.

### See also:

The *command reference* for details.

If you are updating from Turbo Router 2.x, move to the *installation section*. Else, move to the next section.

## Update to the new version

### 1. Import the new image

```
vrouter> cmd system-image import <image-url>
Checking startup configuration compatibility with imported image ...
OK: startup configuration is compatible with version 3.0.0
```

---

**Note:** API deprecation notices may be displayed:

- *deprecated* APIs can safely be fixed after booting with the new version.
  - *obsolete* APIs should be fixed at that step, or the *startup* configuration will be partially applied. Fix the *startup* configuration, delete the new image and restart the update procedure.
- 

### 2. Reboot on the new image

The new image will be booted automatically on next reboot:

```
vrouter> cmd system-image list
3.0.0 (default) (current)
3.0.1 (next)
```

Reboot the appliance:

```
vrouter> cmd reboot delay 0
```

### 3. Set the new image as the default image

After a successful reboot, you can set the new image as default:

```
vrouter> cmd system-image set-default 3.0.1
```

The old image can be deleted with:

```
vrouter> cmd system-image delete 3.0.0
```

**See also:**

The *user guide* and *command reference* for details.

## API deprecation

When the API changes between Turbo Router versions, the new API coexists with the previous one, which is progressively deprecated, obsoleted and removed, as follows:

| GA version | API state  |
|------------|--|
| N          | API v1 is valid and fully supported.   |
| N + 1      | API v2 is introduced; API v1 is <i>deprecated</i> and this is reported at update and when editing the configuration in the CLI. Both APIs are supported; switching to v2 is recommended. |
| N + 2      | API v1 is <i>obsolete</i> . It can still be loaded and displayed in the CLI, but it is ineffective. API v1 is not supported; switching to v2 is required.                                |
| N + 3      | API v1 is <i>removed</i> and cannot be used.   |

## 2.3 First configuration

### 2.3.1 Logging in to the CLI

Log in as `admin` to access the CLI:

```
login: admin
Password: admin
vrouter>
```

**Warning:** For security reasons, it is recommended to change the default passwords of preconfigured users. See *Changing Passwords* for more information about user accounts.

### 2.3.2 Restoring from a backup file

If not done during the installation, you can restore the configurations, private keys, certificates and licenses from a backup file.

The backup file must have been generated on the same or previous minor version (e.g. a backup from 3.0.1 can be restored on 3.0.x or 3.1.x).

To import a backup file:

```
vrouter> cmd backup import url <backup-url>
Checking imported startup configuration compatibility with current image...
```

(continues on next page)

(continued from previous page)

```
OK: startup configuration is compatible with version 3.0.0.  
Please reboot to take changes in account.
```

Then, reboot the Turbo Router:

```
vrouter> cmd reboot delay 0
```

**See also:**

The *command reference* for details.

## 2.3.3 Day-1 configuration

### Automatic Day-1 configuration

Turbo Router includes a Day-1 configuration mechanism that starts a DHCP client on the first interface and enables a SSH server on it, so that the user can remotely access the console.

1. Check the VRF main state:

```
vrouter> show state vrf main  
vrf main  
  (...)  
  interface  
    physical ens3  
      oper-status UP  
      ipv4  
        address 10.0.2.15/24  
        ..  
  (...)  
  ssh-server  
    port 22  
    enabled true
```

Here, we see that the `ens3` interface in the main VRF is configured with an IP address and that SSH is enabled. You can jump to *Configuring the fast path*. If the automatic Day-1 configuration doesn't match your needs, you can perform manual Day-1 configuration:

## Manual Day-1 configuration with static IP address

To configure an address on the management interface and enable SSH from the CLI, proceed as follows:

1. Start to edit the running configuration:

```
vrouter> edit running
vrouter running config#
```

2. Create an interface named eth0 on top of the pci-b0s3 port, in the main vrf:

```
vrouter running config# vrf main interface physical eth0
vrouter running physical eth0#! port pci-b0s3
vrouter running physical eth0# commit
```

---

**Note:** use `show state / network-port` to see the list of available network ports with PCI ids; it can help choosing the right management port.

---

3. Add an address to the management interface and apply the changes:

```
vrouter running physical eth0# ipv4 address 192.168.0.2/24
vrouter running physical eth0# commit
```

4. Check that the system state for the new interface is correct:

```
vrouter running physical eth0# show state
physical eth0
  oper-status UP
  enabled true
  mtu 1500
  ipv4
    address 192.168.0.2/24
    (...)
  port pci-b0s3
  (...)
```

5. Add a default route:

```
vrouter running physical eth0# / vrf main routing static
vrouter running static# ipv4-route 0.0.0.0/0 next-hop 192.168.0.1
vrouter running static# commit
```

6. Enable SSH server:

```
vrouter running static# / vrf main ssh-server
vrouter running ssh-server# commit
vrouter running ssh-server# exit
```

Now the equipment can be accessed via a remote SSH client at address 192.168.0.2.

7. To make this configuration applied at each startup, make it the startup configuration:

```
vrouter> copy running startup
Overwrite startup configuration? [y/N] y
```

## Manual Day-1 configuration with DHCP

To configure an address, a default route via DHCP on the management interface, a DNS and enable SSH from the CLI, proceed as follows:

1. Start to edit the running configuration:

```
vrouter> edit running
vrouter running config#
```

2. Create an interface named eth0 on top of the pci-b0s3 port, in the main vrf:

```
vrouter running config# vrf main interface physical eth0
vrouter running physical eth0#! port pci-b0s3
vrouter running physical eth0# commit
```

---

**Note:** use `show state / network-port` to see the list of available network ports with PCI ids; it can help choosing the right management port.

---

3. Enable DHCP on the management interface and apply the changes:

```
vrouter running physical eth0# ipv4 dhcp
vrouter running dhcp# commit
```

4. Check that the system state for the new interface is correct:

```
vrouter running physical eth0# show state
physical eth0
  (...)
  ipv4
    dhcp
      dhcp-lease-time 7200
      enabled true
```

(continues on next page)

(continued from previous page)

```
current-lease
  renew 3 2018/07/04 04:04:15
  fixed-address 10.0.2.15
  expire 3 2018/07/04 16:26:02
  rebind 3 2018/07/04 13:26:02
  (...)
address 10.0.2.15/24
  (...)
port pci-b0s3
  (...)
```

#### 5. Configure the DNS:

```
vrouter running physical eth0# / vrf main dns server 1.1.1.1
vrouter running physical eth0# commit
```

#### 6. Enable the SSH server:

```
vrouter running physical eth0# / vrf main ssh-server
vrouter running ssh-server# commit
vrouter running ssh-server# exit
```

Now the equipment can be accessed via a remote SSH client using the address acquired by DHCP (in our case 10.0.2.15).

#### 7. To make this configuration applied at each startup, make it the startup configuration:

```
vrouter> copy running startup
Overwrite startup configuration? [y/N] y
```

## 2.3.4 Configuring your license

A license key is required to unlock Turbo Router features and capacities. Refer to the *User Guide* for detailed information about the Turbo Router licensing mechanisms.

### Installing your license

#### Online license

The standard online license requires DNS and internet access and is configured as follows, using the serial number provided with the software delivery:

```
vrouter> edit running
vrouter running config# system license online serial <serial>
vrouter running config# commit
vrouter running config# exit
vrouter>
```

---

**Note:** It is required to exit edit mode to take license activation into account.

---

---

**Note:** If your configuration relies on a feature enabled by license (BGP, IPSEC) to reach the license key server and you have not obtained a valid license yet, you are facing a *catch-22* ([https://en.wikipedia.org/wiki/Catch-22\\_\(logic\)](https://en.wikipedia.org/wiki/Catch-22_(logic))) issue. You'll need to perform a simple Day-1 configuration using static routing to install your license key first, and then you'll be able to move forward with a more complex configuration.

---

### Offline activation of an online license

Some specific licenses can be activated offline, through a webpage on the *Licensing End-User Portal*.

First, request an activation certificate on the machine that will use the license token.

```
vrouter> cmd license certificate request-activation serial <serial>
Activation certificate: <certificate>
```

---

**Note:** The license certificate request commands can only be run when the license is disabled in configuration.

---

Then login to the *Licensing End-User Portal* using the credential received from 6WIND support team, go to the **Offline Activation** tab, copy paste the certificate obtained at the previous step, and click on **Activate**.

---

**Note:** A license token can be freed using `cmd license certificate request-deactivation serial <serial>`, copy paste the obtained certificate to the Licensing End-User Portal, and click on **Deactivate**.

---

The next step is to import this new certificate back on the machine that will use the license token.

```
vrouter> cmd license certificate import content <certificate-from-webpage> serial
↵<serial>
OK.
```

---

**Note:** The import command accepts other url types, like ftp and http. The file contents can be directly pasted on

the console too. Run `cmd license certificate import <?>` for a complete list.

---

Finally, the system can be configured to use the license:

```
vrouter> edit running
vrouter running config# system license offline-certificate serial <serial>
vrouter running config# commit
vrouter running config# exit
vrouter>
```

---

**Note:** It is required to exit edit mode to take license activation into account.

---

## Offline license

In some cases, an offline license file can be provided in addition to your serial number. It must be imported before entering the serial number, as follows:

```
vrouter> cmd license file import url http://path/to/file serial <serial>
vrouter> edit running
vrouter running config# system license offline serial <serial>
vrouter running config# commit
vrouter running config# exit
vrouter>
```

---

**Note:** It is required to exit edit mode to take license activation into account.

---

---

**Note:** The import command accepts other url types, like ftp and http. The file contents can be directly pasted on the console too. Run `cmd license file import <?>` for a complete list.

---

## Showing license information

Use the `show license` command to display the license status.

```
vrouter> show license
Active perpetual license for Turbo Router
Current activations 88/200
Connected to license server (last contact 2020-04-22 21:25:00)
Lease is valid until 2020-05-22 21:24:59
```

(continues on next page)

(continued from previous page)

```

Serial number is xxxxxxxxxxxxxxxx
Computer ID is mLgk6tN3nKRB5Z9Jp1su
License was activated online
Support is valid until 2020-12-31 06:00:00 (standard mode)
Max throughput 100.0G (moving average 0.0G)
IPsec activated for 1000000 tunnels (currently used 0)
CG-NAT activated for 300000000 contracks (currently used 0)
vrouter>

```

You will want to look for the following output to confirm that your license key is active and valid:

- Active perpetual license for Turbo Router
- Connected to license server ...
- Lease is valid until ...
- License was activated online

In case of doubt, jump to the *next section* to learn how to check the license service logs.

The state of the license can also be retrieved using `show state / system license`.

```

vrouter> show state / system license
license
  enabled true
  valid true
  state "Concurrent License Activated"
  activation-type "License was activated online"
  license-type "Perpetual, concurrent"
  short-license-type perpetual
  support-type standard
  support-end-date "2020-12-31 06:00:00"
  throughput
    allowed 100.0
    used 0.0
    ..
  cgnat-contracks
    allowed 300000000
    used 0.0
    ..
  ipsec-tunnels
    allowed 1000000
    used 0.0
    ..
  online
    serial xxxxxxxxxxxxxxxx

```

(continues on next page)

(continued from previous page)

```
connected true
current-activations 88
allowed-activations 200
computer-id mLgk6tN3nKRB5Z9Jp1su
..
..
vrouter>
```

### 2.3.5 Configuring the fast path

The fast path is the Turbo Router component in charge of packet processing. To accelerate ethernet NICs, they must be dedicated to the fast path, and the fast path must be started.

1. Dedicate ports to the fast path and start it:

```
vrouter> edit running
vrouter running config# system fast-path
vrouter running fast-path#! port pci-b0s4
vrouter running fast-path# port pci-b0s5
vrouter running fast-path# show config
fast-path
  enabled true
  port pci-b0s4
  port pci-b0s5
vrouter running fast-path# commit
```

---

**Note:** use `show state / network-port` to see the list of available network ports with PCI ids; it can help choosing the right ports.

---

2. Check that the fast path has been started (it can take some time):

```
vrouter running fast-path# show state
fast-path
  port pci-b0s5
  port pci-b0s4
  enabled true
```

## 2.3.6 Configuring networking

Now that the fast path has been started and some ports have been dedicated to it, we can start the networking configuration.

Let's create the the dp0 and dp1 interfaces in the main VRF and associate them to these two ports. The 1.0.0.1/24 address will be added to dp0, and 2.0.0.1/24 address will be added to dp1.

```
vrouter> edit running
vrouter running config# vrf main
vrouter running vrf main# interface physical dp0 port pci-b0s4
vrouter running vrf main# interface physical dp0 ipv4 address 1.0.0.1/24
vrouter running vrf main# interface physical dp1 port pci-b0s5
vrouter running vrf main# interface physical dp1 ipv4 address 2.0.0.1/24
vrouter running vrf main# commit
```

## 2.4 Advanced Features

### 2.4.1 Automated pre-configuration using Cloud-init

If you installed Turbo Router as a new Linux system, it includes a Day-1 configuration mechanism that starts a DHCP client on the first interface and enables a SSH server on it, so that the user can remotely access the console. This mechanism relies on cloud-init and can be customized as described in the following sections.

#### Cloud-init

Cloud-init is the defacto multi-distribution package that handles early initialization of a cloud instance. Using cloud-init, it is possible to preconfigure Turbo Router.

#### See also:

For more information about Cloud-init, refer to <https://cloudinit.readthedocs.io/en/latest/>

Customizing the Turbo Router configuration files is possible only at first boot. The turbo service is started sooner in the next boots, before cloud-init.

#### Libvirt

The simpler way of using cloud-init with libvirt is to create an iso file labelled cidata.

#### See also:

For more information, refer to <https://cloudinit.readthedocs.io/en/latest/topics/datasources/nocloud.html>

1. Write a user-data file and a meta-data file. In this example, we setup the root password.

```
cat << EOF > /tmp/user-data
#cloud-config
chpasswd:
  list: |
    root:myrootpassword
EOF

cat << EOF > meta-data
instance-id: turbo-vm
local-hostname: turbo-vm
EOF
```

2. Build an iso image with the cidata label containing the user-data and meta-data and put it in the libvirt images directory.

```
apt-get install -y genisoimage
genisoimage -output seed.iso -volid cidata \
            -joliet -rock user-data meta-data
cp seed.iso /var/lib/libvirt/images/
```

3. Add seed.iso as a disk to the virt-install command. For instance, for a VM with virtual NICs.

```
virt-install --name vm1 --vcpus=3,sockets=1,cores=3,threads=1 \
            --os-variant ubuntu18.04 --cpu host --network=default,model=e1000 \
            --ram 8192 --noautoconsole --import \
            --disk /var/lib/libvirt/images/vm1.qcow2,device=disk,bus=virtio \
            --disk /var/lib/libvirt/images/seed.iso,device=disk,bus=virtio
```

## OpenStack

Cloud-init is integrated within OpenStack.

1. Write a cloud-init user-data file. In this example, we setup the root password.

```
cat << EOF > /tmp/user-data
#cloud-config
chpasswd:
  list: |
    root:myrootpassword
EOF
```

2. Start the VM with the additional parameter --user-data.

```
openstack server create --flavor turbo-router \
                        --image turbo-router \
```

(continues on next page)

(continued from previous page)

```
--user-data /tmp/user-data \  
turbo-router_vm
```

## Examples

Here is a `user-data` example, where we upload a startup configuration for the CLI (you can also upload alternative configurations).

```
#cloud-config  
write_files:  
- path: /run/vrouter.startup  
  content: |  
    {  
      "vrouter:config": {  
        "vrf": [  
          {  
            "name": "main",  
            "vrouter-interface:interface": {  
              "physical": [  
                {  
                  "name": "pub1",  
                  "port": "pci-b0s5",  
                  "ipv4": {  
                    "dhcp": {  
                      "enabled": true  
                    }  
                  }  
                }  
              ]  
            }  
          ]  
        }  
      },  
      "vrouter-system:system": {  
        "vrouter-fast-path:fast-path": {  
          "port": [  
            "pci-b0s5"  
          ]  
        },  
        "vrouter-license:license": {  
          "online": {  
            "serial": "xxx"  
          }  
        }  
      }  
    }  
  ]
```

(continues on next page)

(continued from previous page)

```
    }  
  }  
}  
runcmd:  
- [ /usr/bin/sysrepcfg, -m, vrouter, -d, startup, -f, json, --import=/run/vrouter.  
↪ startup ]  
- [ /usr/bin/sysrepcfg, -m, vrouter, -C, startup ]
```

## 3. User Guide

### 3.1 User Guide - CLI / NETCONF

6WIND command line interface (CLI) is the user interface to interact with a device running 6WIND vRouter. It can be used to configure, monitor and troubleshoot the router.

The CLI is a NETCONF client, following a data model described in YANG. The command names and statements follow the syntax and the hierarchical organization of the vRouter YANG models.

A NETCONF API can be used as well from any other NETCONF clients to configure and monitor the router remotely.

#### About NETCONF

NETCONF is a network management protocol standardized by the IETF. It defines mechanisms to install, manipulate and delete the configuration of network devices. It uses Extensible Markup Language (XML)-based data encoding for the configuration data as well as the protocol messages. More information is available in RFC 6241 at <https://tools.ietf.org/html/rfc6241>

#### About YANG

YANG is a language used to model data for the NETCONF protocol. A YANG module defines a hierarchy of data that can be used for NETCONF-based operations, including configuration, state data, Remote Procedure Calls (RPCs), and notifications for network management protocols. More information is available in RFC 7950 <https://tools.ietf.org/html/rfc7950>

This document is organized as follows:

#### 3.1.1 Preface

##### Conventions

In this document, the following conventions are used:

| Convention | Description  |
|------------|--|
| literal    | CLI keywords.  |
| <value>    | CLI arguments for which the user is supposed to supply values. |
| UPPERCASE  | A keyboard key or combination.                                 |
| [X]        | Square brackets indicate optional elements.                    |
| X Y        | A pipe indicates a logical or (exclusive).                     |

## Definitions

**Mode** A mode is an environment providing a list of CLI commands. The operational mode mostly provides commands to display state, while the edition mode provides commands to modify the device configuration.

**Context** A context is an environment of the edition mode in which parameters can be configured or displayed. Some CLI commands are relevant to a context.

**Configuration** A configuration describes a coherent programming of the device, represented in a tree.

**Staging Configuration** The staging configuration is the one currently being modified locally in the CLI, and not yet active on the device.

**Running Configuration** The running configuration is the one currently active on the device.

**Startup Configuration** The startup configuration is the one that will be loaded at the next reboot.

**Configuration File** A configuration file is used to transfer a configuration to or from a remote machine for editing or backup purposes.

### 3.1.2 Key features

The key features of the CLI are:

- *Command line*
- *NETCONF API*
- *Clear separation between configuration and state data*
- *Multiple logical VRF*
- *Compatibility with Day-1 configuration*

## Command line

The CLI comes with traditional features, such as completion, history and contextual help. It relies on a YANG data model that users browse as they would browse a file system, for example, `/` jumps to the root of the configuration, `..` moves one level up. Relative and absolute paths can be used to refer to configuration data, making browsing very efficient.

## NETCONF API

The management system embeds a NETCONF server which can be configured to accept external connections from a NETCONF client. It supports all the required protocol operations to read and write the configuration: `<get>`, `<get-config>`, `<edit-config>`, `<copy-config>` and so on.

The CLI is actually a client that connects locally to this NETCONF server.

## Clear separation between configuration and state data

At the root of the data model, there are two trees: `config` and `state`. The items in `config` represent the target configuration, while the ones in `state` represent the actual state of the system. As a result, `state` generally includes the items in `config`, plus additional runtime information, such as the statistics, or the IP addresses obtained through DHCP for example.

## Multiple logical VRF

The management system splits the device into VRFs. Each VRF has its own set of IP addresses, routing tables, firewall rules, and other network-related resources. The configuration of most networking services occurs inside a VRF context. The default VRF is called `main`.

VRFs rely on the Linux network namespaces feature (`netns`). This kind of container may be used in future releases to define limits in term of CPU resource or memory.

## Compatibility with Day-1 configuration

Cloud-init is embedded in the vRouter for Day-1 configuration, that is, the initial configuration of the vRouter to enable basic console access. By default, cloud-init starts a DHCP client on the first interface and enables a SSH server on it. It can be customized to configure a specific interface, use a static IP address, create users, provision SSH keys, etc.

The management engine is compatible with cloud-init Day-1 configuration, as it does not touch the network services (SSH, DNS, DHCP, etc.) as long as there is no configuration statement for them. When a configuration statement is present, it takes precedence over any existing configuration coming from cloud-init. Finally, a known service like SSH will be recognized and will not be restarted if it is not necessary.

### 3.1.3 Basics

#### Overview

The CLI is used to configure the device and monitor its state. The CLI exposes two main modes:

- The operational mode (prompt is `hostname>`), where the user can query the state, show the configuration, send commands, etc...
- The edition mode (prompt ends with `#`), where the user can additionally modify the configuration of the device. This mode is divided into several service contexts, each of them representing a part of the configuration.

At any level in the CLI, if you type a question mark (`?`), a list of available commands and a short help is displayed.

The CLI connects to the device using the NETCONF protocol. The contexts commands are generated from YANG files, which also describes the NETCONF API of the device.

Here is a summary of available commands from the CLI prompt:

|  |  |
|--|--|
| <code>?</code>   | Display contextual help.   |
| <code>help [&lt;command&gt;]</code>                                    | Display the commands list or the help of a command.  |
| <code>TAB</code>   | Complete or display options from current line.   |
| <code>..</code>  | Move one level up to parent context.   |
| <code>save file &lt;name&gt;</code>                                    | Save staging configuration to a file.  |
| <code>load file &lt;name&gt;</code>                                    | Load a file into the staging configuration.  |
| <code>commit</code>  | Commit pending changes in the running.   |
| <code>show state [&lt;path&gt;]</code>                                 | Fetch the state of the device.   |
| <code>show config [&lt;confname&gt;] [&lt;path&gt;]</code>             | Show the configuration (staging configuration by default in edition mode, or running configuration in operational mode). |
| <code>diff [&lt;confname1&gt; &lt;confname2&gt;] [&lt;path&gt;]</code> | Show the differences between two configurations.   |
| <code>exit</code> or <code>CTRL-D</code>                               | Exit from edition mode, or exit the CLI when in operational mode.  |
| <code>UP, DOWN</code>  | Browse the command history.  |
| <code>#</code>   | At the beginning of the line or after a space, all characters after the <code>#</code> are interpreted as comments.      |

The CLI stores the history of typed commands in a circular memory. Typed commands can be recalled with the UP/DOWN keys and may be modified with LEFT/RIGHT/INS/DEL keys.

## Output modifiers

The cli commands can be suffixed by output modifiers to change the output of the command. The syntax is quite similar to shell pipes.

The pager can be disabled for a specific command with:

```
vrouter> show state | no-pager  
(...)
```

Similarly, a pager modifier can force the activation of the pager.

The match output modifier acts like the grep shell command. Its syntax is match <pattern> [invert] [count] [context <n>], with:

- <pattern>: a regular expression
- invert: invert the pattern
- count: count the number of matches
- context <n>: show n lines before and after each match

For instance, this command counts the number of physical interfaces whose mtu is 1500:

```
vrouter> show state fullpath | match 'interface physical' | match 'mtu 1500' count  
3
```

## Logging in to the CLI

Log in as admin to access the CLI:

```
login: admin  
Password: admin  
vrouter>
```

**Warning:** For security reasons, it is recommended to change the default passwords of preconfigured users.  
See *Changing Passwords* for more information about user accounts.



(continued from previous page)

This command supports several output formats, and can be constrained to a specific path.

Command syntax: `show config [staging|running|startup|(file <file>)] \ [text|xml|json] [all|nodefault] [relative|absolute] \ [fullpath|nopath] [<path...>]`

`show state:`

Show the system state.

In edition mode, show the state of the current path.

In operational mode, show the full the state of the system.

This command supports several output formats, and can be constrained to a specific path.

Command syntax: `show state [text|xml|json] [all|nodefault] [relative|absolute] \ [fullpath|nopath] [<path...>]`

`show <service>`

Show a service configuration.

Command syntax: `show [dry-run] [text|xml|json] <service> [args...]`

`vrouter running physical eth0# help show`

The context-sensitive help can be requested at any time by entering a question mark ?. It displays a list of available options:

```
vrouter running physical eth0# i?
  ipv4          Parameters for the IPv4 address family.
  ipv6          Parameters for the IPv6 address family.
vrouter running physical eth0# ipv4 ?
<return>      Validate command.
address       The list of configured IPv4 addresses on the interface.
dhcp         DHCP client configuration.
(...)
```

## Operational mode

When connecting to the CLI, it starts in operational mode, identified by the following prompt:

```
vrouter>
```

In this mode, the user can:

- display the help of a command (ex: `help edit`)
- retrieve the state of the device (ex: `show state`)
- retrieve the running configuration of the device (ex: `show config`)
- switch to the edition mode (ex: `edit running`)

- update the startup configuration of the device (copy running startup)
- send commands (ex: cmd reboot)

## Show configuration

The `show config` command is used to display the configuration. In operational mode, it shows the running configuration by default.

The syntax of the command is: `show config [running|startup|(file <file>)] [text|xml|json] [all|nodefault] [relative|absolute] [fullpath|nopath] [<path...>]`

The default output format is text:

```
vrouter> show config /
config
  vrf main
    ssh-server
      enabled true
      port 22
      ..
  ..
  ..
```

The output format can be customized. See the *Edition Mode section* for details.

## Show state

The `show state` command is used to display the current state of the device. The arguments and the output of the command are similar to the `show config` command.

The syntax of the command is `show state [text|xml|json] [all|nodefault] [relative|absolute|fullpath] [<path...>]`.

Example of use:

```
vrouter> show state network-port
network-port pci-b0s4
  pci-bus-addr 0000:00:04.0
  vendor "Red Hat, Inc."
  model "Virtio network device"
  mac-address 52:54:00:12:34:57
  interface eth0
  ..
network-port pci-b0s3
  pci-bus-addr 0000:00:03.0
```

(continues on next page)

(continued from previous page)

```

vendor "Red Hat, Inc."
model "Virtio network device"
mac-address de:ad:de:01:02:03
interface eth1
..
network-port pci-b0s2
pci-bus-addr 0000:00:02.0
vendor "Red Hat, Inc."
model "Virtio network device"
mac-address 52:54:00:12:34:56
interface eth2
..

```

## Diff configurations

The `diff` command shows the differences between two configurations. The syntax is the same than in edition mode, except that the configurations to be diffed must always be specified.

See the *Edition Mode section* for details.

## Showing the state summary of the device

The CLI is able to display the state summary of the device.

This operation is invoked with the `show summary` CLI command, which is available from the operation mode and from the edition mode.

Show the summary of the device:

```

vrouter> show summary
Service                Status
=====                =====

license                enabled, valid
product                Turbo Router 3.0.0

fast-path              disabled
linux                  3 cores, memory total 0.96GB available 0.09GB
network-port           1 port detected

vrf                    1 configured

interface physical     enabled in vrf main (1 up iface)

```

(continues on next page)

(continued from previous page)

```

interface system-loopback enabled in vrf main (1 up iface)

routing                enabled in vrf main (2 ipv4 routes, 2 ipv6 routes)

auth                   2 users
cloud-init             enabled
dhcp client            enabled in vrf main
dns                    enabled in vrf main
ssh-server             enabled in vrf main

```

### Showing the operational state of the device

The CLI is able to query the state of the device. What is called *state* is the current operational state of the system, in contrast to *config* which contains the administrative desired state.

This operation is invoked with the `show state` CLI command, which is available from the operation mode and from the edition mode.

Show all the state of the device in text format:

```

vrouters> show state
state
  system
    hostname ubuntu1604
    fast-path
      enabled false
    ..
(...)

```

In edition mode, the displayed state corresponds to the service context being edited.

Show the interfaces of the device in text format:

```

vrouters> edit running
vrouters running config# vrf main interface
vrouters running interface# show state
interface
  physical ens2
  oper-status DOWN
  ethernet
    mac-address 52:54:00:12:34:56
    ..
(...)

```

By default, the output is in text format. But it can also be displayed in xml or json.

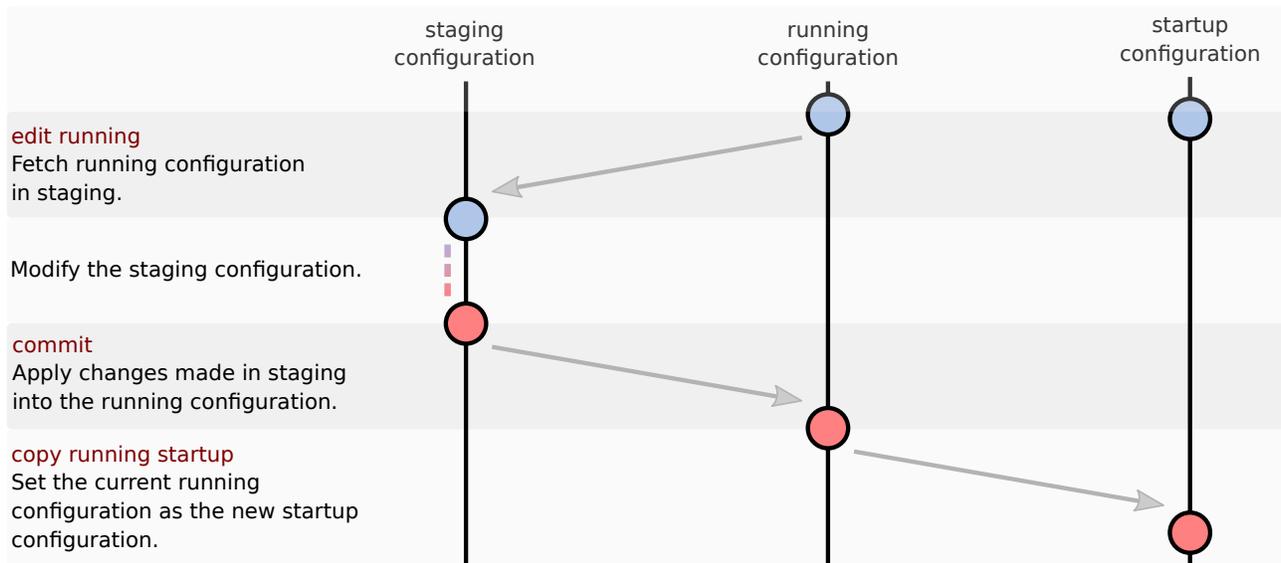
Show the interfaces of the device in xml format:

```
vrouters> show state
vrouters running config# vrf main interface
vrouters running interface# show state xml
<interface xmlns="urn:6wind:vrouter/interface">
  <physical>
    <name>ens2</name>
  </interface>
  (...)

```

## Editing the running configuration

The edition model is transactional. The running configuration is first fetched locally. This local copy, called *staging configuration*, can be modified locally, then committed. The running configuration can be set as startup configuration.



Enter into the edition mode with:

```
vrouters> edit running
vrouters running config#

```

In edition mode, the prompt is composed of:

- the hostname,
- the name of the configuration being edited (here `running`),
- the path of the current node in the configuration tree (here `/`, which means we are at the root),
- a `#`, meaning we are in the edition mode.

A ! can also be displayed at the end of the prompt when the staging configuration is invalid regarding the constraints defined in the YANG model. The `validate` command can then be used to check what is invalid in the configuration:

```
vrouter running interface# physical eth1
vrouter running physical eth1#! validate
ERR ly Missing required element "port" in "physical".
Invalid configuration.
vrouter running physical eth1#! port pci-b0s2
vrouter running physical eth1#
```

In edition mode, the user can:

- modify the staging configuration (ex: `vrf main ssh-server`)
- show the staging configuration (ex: `show config`)
- commit the changes (ex: `commit`)
- discard the changes (ex: `exit`)
- display the help of a command (ex: `help show`)
- retrieve the state of the device (ex: `show state`)
- update the startup configuration of the device (`copy running startup`)
- send commands (ex: `cmd reboot`)

## Edition mode

### Enter into a context

The configuration is organized hierarchically. All configuration is available under the `config` node.

```
config/
├── system
│   ├── auth
│   ├── fast-path
│   └── ...
├── vrf
│   ├── dns
│   ├── interface
│   └── ...
```

To enter into a context, type its name, followed by the key in case of a list.

```
vrouter running config#
vrouter running config# vrf main
vrouter running vrf main# interface
```

(continues on next page)

(continued from previous page)

```
vrouter running interface# physical eth0
vrouter running physical eth0#
```

This can also be done in one command:

```
vrouter running config# vrf main interface physical eth0
vrouter running physical eth0#
```

---

**Note:** The CLI commands are generated from YANG files, which also specifies the NETCONF API of the device. A CLI context corresponds to a *container* or a *list* statement in the YANG file.

---

## Set configuration values

To set the value of a leaf, type its name and its value:

```
vrouter running physical eth0# port pci-b0s4
vrouter running physical eth0# mtu 1500
vrouter running physical eth0# show config
physical eth0
  (...)
  port pci-b0s4
  mtu 1500
  (...)
```

Several leaves can be set in one command, achieving the same result:

```
vrouter running physical eth0# port pci-b0s4 mtu 1500
vrouter running physical eth0#
```

Finally, it is possible to set the value of leaves that are in a different path. In that case, specify the path, followed by the leaves and their values. Note that the current directory remains unchanged.

```
vrouter running config# vrf main interface physical eth0 mtu 1500 port pci-b0s4
vrouter running config#
```

---

**Note:** The CLI commands are generated from YANG files, which also specifies the NETCONF API of the device. A CLI configuration leaf corresponds to a *leaf* or a *leaflist* statement in the YANG file.

---

## Delete a configuration node

A configuration node (either a leaf or a context) can be deleted with the command `del`, followed by the path of the node:

```
vrouter running physical eth0# mtu 1500
vrouter running physical eth0# show config
physical eth0
  (...)
  mtu 1500
  (...)
vrouter running physical eth0# del mtu
vrouter running physical eth0# show config
[... no mtu ...]
```

## Complex configuration commands

Some commands need to have a more complex syntax, because a couple name/value is not sufficient. In this case, the CLI behavior is customized with extensions in the YANG files.

Particularly, a YANG *container* or *list* can be used to define *oneline* commands. For example, the interface IP neighbor context uses an extension to have a specific syntax:

```
neighbor <ip> link-layer-address <mac>
```

The following example shows that it does not follow the same syntax than the simple case described above. Each neighbor is identified by its key, and the argument attached to the neighbor is mandatory. To delete a neighbor, only the key is needed.

```
vrouter running ipv4# neighbor 10.100.0.0 link-layer-address 11:11:11:11:11:11
vrouter running ipv4# neighbor 10.200.0.0 link-layer-address 22:22:22:22:22:22
vrouter running ipv4# show config
ipv4
  neighbor 10.100.0.0 link-layer-address 11:11:11:11:11:11
  neighbor 10.200.0.0 link-layer-address 22:22:22:22:22:22
  enabled true
  ..
vrouter running ipv4# del neighbor 10.100.0.0
vrouter running ipv4# show config
ipv4
  neighbor 10.200.0.0 link-layer-address 22:22:22:22:22:22
  enabled true
  ..
```

## Show configuration

The `show config` command is used to display the configuration. In edition mode, it shows the staging configuration by default, relative to the current path.

The syntax of the command is: `show config [staging|running|startup|(file <file>)] [text|xml|json] [all|nodefault] [relative|absolute] [fullpath|nopath] [<path...>]`

---

**Note:** `show config` (show the configuration) should not be confused with `show state` (get the operational state).

---

```
vrouter running config# vrf main ssh-server
vrouter running ssh-server# show config
ssh-server
  enabled true
  port 22
  ..
```

It is possible to show the running or the startup configuration:

```
vrouter running config# vrf main ssh-server
vrouter running ssh-server# show config running
ssh-server
  enabled true
  port 22
  ..
```

The configuration can be displayed in different format (text, xml or json):

```
vrouter running config# vrf main ssh-server
vrouter running ssh-server# show config json
{
  "vrouter-ssh-server:ssh-server": {
    "enabled": true,
    "port": 22
  }
}
```

The configuration nodes set to the default value can be stripped from the configuration with `nodefault` (in this example port set to 22 and enabled set to true are not displayed):

```
vrouter running config# vrf main ssh-server
vrouter running ssh-server# show config xml nodefault
<ssh-server xmlns="urn:6wind:vrouter/ssh-server">
</ssh-server>
```

A path can be specified, which can be absolute, or relative to the current path:

```

vrouter running config# vrf main ssh-server
vrouter running ssh-server# show config
ssh-server
  enabled true
  port 22
  ..
vrouter running ssh-server#
vrouter running ssh-server# show config .. ..
config
  vrf main
    ssh-server
      enabled true
      port 22
      ..
    ..
  ..
vrouter running ssh-server# show config /
config
  vrf main
    ssh-server
      enabled true
      port 22
      ..
    ..
  ..
vrouter running ssh-server# show config / vrf main ssh-server
ssh-server
  enabled true
  port 22
  ..

```

The configuration root path can be relative (default), or absolute. If **absolute** is specified, all the parent containers are displayed, but the configuration that is not in the specified path is stripped. This example demonstrates the feature:

```

vrouter running ssh-server# show config /
vrf main
  ssh-server
    enabled true
    port 22
    ..
  ..
vrf vr1

```

(continues on next page)

(continued from previous page)

```

..
vrouters running ssh-server# show config
ssh-server
  enabled true
  port 22
..
vrouters running ssh-server# show config absolute
vrf main
  ssh-server
    enabled true
    port 22
  ..
..

```

When the configuration is displayed in a text format, the full path can be prepended to each node. This eases copy/paste, or filtering using the match output filter:

```

vrouters running ssh-server# show config fullpath
/ vrf main ssh-server
/ vrf main ssh-server enabled true
/ vrf main ssh-server port 22

```

The `show config` command is also available from the operational mode. In this case, the running configuration is displayed by default as there is no staging configuration.

## Show state

The `show state` command is used to display the current state of the device. The arguments and the output of the command are similar to the `show config` command.

The syntax of the command is `show state [text|xml|json] [all|nodefault] [relative|absolute|fullpath] [<path...>]`.

Without path argument, the displayed state depends on the current location in the configuration. At root, it displays all the state:

```

vrouters running config# show state
vrf main
  network-stack
    icmp
      ignore-icmp-echo-broadcast false
      rate-limit-icmp 1000
      rate-mask-icmp destination-unreachable source-quench time-exceeded
↪parameter-problem

```

(continues on next page)

(continued from previous page)

```

    ..
    ipv4
        forwarding true
(...)

```

When called from an interface context, only the state of this interface is displayed:

```

vrouter running physical ens2# pwd
/ vrf main interface physical ens2
vrouter running physical ens2# show state
physical ens2
    mtu 1500
    promiscuous false
    enabled false
    port pci-b0s2
    rx-cp-protection false
(...)

```

Like in the `show config` command, the path and the output format can be specified.

## Diff configurations

The `diff` command shows the differences between two configurations. Additions are prefixed by a `+` and deletions by a `-`. All lines changed in the same directory are prefixed by a title line starting with `===`.

Without argument, it displays the differences between the origin configuration and the staging configuration in the current directory: in other words, it shows the uncommitted user changes.

```

vrouter running config# vrf main
vrouter running vrf main# interface physical eth0
vrouter running physical eth0#! port pci-b0s2
vrouter running physical eth0# diff
=== / vrf main interface
+ physical eth0
+     port pci-b0s2
+     enabled true
+     ipv4
+         enabled true
+         ..
+     ipv6
+         enabled true
+     ..
+     ..

```

A path argument can be appended:

```

vrouter running physical eth0# diff /
=== /
+ vrf main
+   interface
+     physical eth0
+       port pci-b0s2
+       enabled true
+       ipv4
+         enabled true
+         ..
+       ipv6
+         enabled true
+         ..
+     ..
+ ..
vrouter running physical eth0# diff ..

```

The configurations used for the diff can be specified:

```

vrouter running fast-path# diff file my-config startup
=== / system
- fast-path
-   enabled false
-   port pci-b0s2
-   cp-protection
-     budget 10
-     ..
-   linux-sync
-     fpm-socket-size 2097152
-     nl-socket-size 67108864
-     ..
- ..

```

If the fullpath argument is passed, each line is expressed with an absolute path:

```

vrouter running config# diff fullpath running staging /
=== /
+ / vrf vr0
+ / vrf vr0 interface
+ / vrf vr0 interface loopback loop0
+ / vrf vr0 interface loopback loop0 enabled true
+ / vrf vr0 interface loopback loop0 ipv4
+ / vrf vr0 interface loopback loop0 ipv4 enabled true
+ / vrf vr0 interface loopback loop0 ipv6

```

(continues on next page)

(continued from previous page)

```
+ / vrf vr0 interface loopback loop0 ipv6 enabled true
=== / system fast-path
- / system fast-path enabled true
+ / system fast-path enabled false
```

## Commit configuration changes

Once you are satisfied with your changes in the staging configuration, you can apply the changes by committing the configuration. This operation copies the content of the staging configuration into the running configuration.

```
vrouter> edit running
vrouter running config# vrf main
vrouter running vrf main# ssh-server
vrouter running ssh-server# show config
ssh-server
  enabled true
  port 22
  ..
vrouter running ssh-server# show config running
vrouter running ssh-server# commit
Configuration committed.
vrouter running ssh-server# show config running
ssh-server
  enabled true
  port 22
  ..
```

---

**Note:** After a call to `commit`, the running configuration is updated immediately. In contrast, the state of the system can take some time to change, depending on the configuration.

---

## Clear configuration changes

Exiting the edition mode cancels the changes done in the staging configuration.

```
vrouter running config# exit
Exit: not saved/applied, are you sure? [y/N] y
```

## Setting the startup configuration

The *startup* configuration is the configuration applied when the device boots. It can be copied from the running configuration, using the following command:

```
vrouter> copy running startup
Overwrite startup configuration? [y/N] y
```

## Handling inactive configuration files

### Save a configuration

In edition mode, the save command can export the staging configuration in a `xml` file.

```
vrouter running config# save file config.xml
Saving in /home/admin/.config/nc-cli/conf/config.xml
vrouter running config#
```

### Load a configuration

In edition mode, the load command sets the staging configuration from a previously saved configuration file.

```
vrouter running config# load file config.xml
Loading a new configuration will overwrite current.
Are you sure? [y/N] y
Loading configuration /home/admin/.config/nc-cli/conf/config.xml
vrouter running config#
```

The staging configuration can also be set to the startup configuration with the following command:

```
vrouter running config# load startup
Loading a new configuration will overwrite current.
Are you sure? [y/N] y
Loading configuration startup
```

## Copying and removing configurations

From edition or operational mode, it is possible to copy and remove configurations.

Here are some examples:

```
vrouters> copy running startup
Overwrite startup configuration? [y/N] y
vrouters> copy running file running.xml
vrouters> copy startup file startup.xml
vrouters> copy file config.xml file config2.xml
vrouters> copy xml file config.json file config2.xml

vrouters> remove startup
Definitively remove startup? [y/N] y
ubuntu1804 running config# remove file config.xml
Definitively remove /home/user/.config/nc-cli/conf/config.xml? [y/N] y
```

To be certain that the startup configuration is a valid one, only the running configuration can be copied to startup.

## Importing and exporting configurations

From edition or operational mode, it is possible to import and export configurations:

```
vrouters> import config.xml http://server/path/to/config.xml
vrouters> import config.json https://server/path/to/config.json
vrouters> import config.xml ftp://user:password@server/path/to/config.xml

vrouters> export config.xml ftp://user:password@server/path/to/config.xml
```

### 3.1.4 System

#### License

#### Overview

#### License key

A license key is required to unlock Turbo Router features and capacities, according to the purchased licenses.

| License Type | License Name | Capacity                         |
|--------------|--------------|----------------------------------|
| Network      | Turbo Router | Throughput: 100Mbps to 200 Gbps  |
| Application  | Turbo IPsec  | Number of tunnels: 100 to 40K    |
|              | Turbo CG-NAT | Number of connections: 1M to 30M |

A single license key is used to enable a set of features and capacities. You should have received your license key at time of software delivery. You can retrieve your license key from the *Licensing End-User Portal* accessible on the 6WIND Customer Zone, or request it by filing a ticket on the 6WIND Customer Zone.

### Activation and health check

The license key must first be configured and activated on Turbo Router. This is done through a connection to the license key server (and therefore requires the DNS and an internet route to be configured). Activating a license key consumes an activation on the license key server. The license is valid if it is enabled on the license key server and has enough activations left.

The license key is valid for 30 days. A health check mechanism regularly and automatically checks the license key validity with the license server. A successful health check resets the validity period for 30 more days.

---

**Note:** During activation or health check, the following information is sent to the license server: the instance's IP address and geolocalization, Turbo Router information (OS, version, build) and machine information (CPU, memory, hostname).

---

---

**Note:** Some specific licenses can be activated offline, through the *Licensing End-User Portal* accessible on the 6WIND Customer Zone. In this case the license can be activated/deactivated through a webpage, and does not require a connection between the Turbo Router and the license server.

---

---

**Note:** In some cases, offline licenses may be provided, which are permanently activated and do not require a connection to the license server.

---

When the license key is removed from the configuration or deactivated on the server, it becomes invalid on the vRouter<sup>1</sup>. The features and capacities will be restricted.

### Maintenance end date

The license key has a maintenance end date associated, which corresponds to the end of the maintenance period after the first delivery of Turbo Router. The maintenance end date is updated when the maintenance agreement is renewed. The license key is not valid for releases of Turbo Router released after the maintenance end date.

---

<sup>1</sup> The license key becomes invalid either right away when removed from the configuration, or at next health check when deactivated on the server.

## Installing your license

### Online license

The standard online license requires DNS and internet access and is configured as follows, using the serial number provided with the software delivery:

```
vrouter> edit running
vrouter running config# system license online serial <serial>
vrouter running config# commit
vrouter running config# exit
vrouter>
```

---

**Note:** It is required to exit edit mode to take license activation into account.

---

**Note:** If your configuration relies on a feature enabled by license (BGP, IPSEC) to reach the license key server and you have not obtained a valid license yet, you are facing a *catch-22* ([https://en.wikipedia.org/wiki/Catch-22\\_\(logic\)](https://en.wikipedia.org/wiki/Catch-22_(logic))) issue. You'll need to perform a simple Day-1 configuration using static routing to install your license key first, and then you'll be able to move forward with a more complex configuration.

---

### Offline activation of an online license

Some specific licenses can be activated offline, through a webpage on the *Licensing End-User Portal*.

First, request an activation certificate on the machine that will use the license token.

```
vrouter> cmd license certificate request-activation serial <serial>
Activation certificate: <certificate>
```

---

**Note:** The license certificate request commands can only be run when the license is disabled in configuration.

---

Then login to the *Licensing End-User Portal* using the credential received from 6WIND support team, go to the **Offline Activation** tab, copy paste the certificate obtained at the previous step, and click on **Activate**.

---

**Note:** A license token can be freed using `cmd license certificate request-deactivation serial <serial>`, copy paste the obtained certificate to the Licensing End-User Portal, and click on **Deactivate**.

---

The next step is to import this new certificate back on the machine that will use the license token.

```
vrouter> cmd license certificate import content <certificate-from-webpage> serial  
↪<serial>  
OK.
```

---

**Note:** The import command accepts other url types, like ftp and http. The file contents can be directly pasted on the console too. Run `cmd license certificate import <?>` for a complete list.

---

Finally, the system can be configured to use the license:

```
vrouter> edit running  
vrouter running config# system license offline-certificate serial <serial>  
vrouter running config# commit  
vrouter running config# exit  
vrouter>
```

---

**Note:** It is required to exit edit mode to take license activation into account.

---

## Offline license

In some cases, an offline license file can be provided in addition to your serial number. It must be imported before entering the serial number, as follows:

```
vrouter> cmd license file import url http://path/to/file serial <serial>  
vrouter> edit running  
vrouter running config# system license offline serial <serial>  
vrouter running config# commit  
vrouter running config# exit  
vrouter>
```

---

**Note:** It is required to exit edit mode to take license activation into account.

---

---

**Note:** The import command accepts other url types, like ftp and http. The file contents can be directly pasted on the console too. Run `cmd license file import <?>` for a complete list.

---

## Showing license information

Use the `show license` command to display the license status.

```
vrouter> show license
Active perpetual license for Turbo Router
Current activations 88/200
Connected to license server (last contact 2020-04-22 21:25:00)
Lease is valid until 2020-05-22 21:24:59
Serial number is xxxxxxxxxxxxxxxx
Computer ID is mLgk6tN3nKRB5Z9Jp1su
License was activated online
Support is valid until 2020-12-31 06:00:00 (standard mode)
Max throughput 100.0G (moving average 0.0G)
IPsec activated for 100000 tunnels (currently used 0)
CG-NAT activated for 30000000 contracks (currently used 0)
vrouter>
```

You will want to look for the following output to confirm that your license key is active and valid:

- Active perpetual license for Turbo Router
- Connected to license server ...
- Lease is valid until ...
- License was activated online

In case of doubt, jump to the *next section* to learn how to check the license service logs.

The state of the license can also be retrieved using `show state / system license`.

```
vrouter> show state / system license
license
  enabled true
  valid true
  state "Concurrent License Activated"
  activation-type "License was activated online"
  license-type "Perpetual, concurrent"
  short-license-type perpetual
  support-type standard
  support-end-date "2020-12-31 06:00:00"
  throughput
    allowed 100.0
    used 0.0
    ..
  cgnat-contracks
    allowed 30000000
```

(continues on next page)

(continued from previous page)

```

    used 0.0
    ..
ipsec-tunnels
    allowed 100000
    used 0.0
    ..
online
    serial xxxxxxxxxxxxxxxx
    connected true
    current-activations 88
    allowed-activations 200
    computer-id mLgk6tN3nKRB5Z9Jp1su
    ..
    ..
vrouter>

```

## Checking license information

### Nominal case

The `show log service license` provides the licensing logs, including license activation and health check details.

```

vrouter running license# show log service license
-- Logs begin at Wed 2020-04-22 21:06:14 UTC, end at Thu 2020-04-23 12:17:01 UTC. --
Apr 22 21:24:54 localhost systemd[1]: Starting vRouter License Daemon...
Apr 22 21:24:54 localhost vrlld[3368]: license_code: xxxxxxxxxxxxxxxx
Apr 22 21:24:54 localhost vrlld[3368]: product: turbo-router
Apr 22 21:24:54 localhost vrlld[3368]: workdir: /var/cache/license
Apr 22 21:24:54 localhost vrlld[3368]: Using cloud license server
Apr 22 21:24:54 localhost vrlld[3368]: License status from cache -1
Apr 22 21:24:54 localhost vrlld[3368]: Dumping state to /var/cache/license/state.json
Apr 22 21:24:54 localhost vrlld[3368]: healthcheck period: 21600 seconds
Apr 22 21:24:55 localhost vrlld[3368]: healthcheck: Concurrent License Activated (3)
Apr 22 21:24:55 localhost vrlld[3368]: computer id: mLgk6tN3nKRB5Z9Jp1su
Apr 22 21:24:55 localhost vrlld[3368]: started
Apr 22 21:24:55 localhost systemd[1]: Started vRouter License Daemon.
Apr 22 21:25:00 localhost vrlld[3368]: healthcheck: Concurrent License Activated (3)
Apr 22 21:25:00 localhost vrlld[3368]: computer id: mLgk6tN3nKRB5Z9Jp1su
Apr 23 03:25:00 vrouter vrlld[3368]: healthcheck: Concurrent License Activated (3)
Apr 23 03:25:00 vrouter vrlld[3368]: computer id: mLgk6tN3nKRB5Z9Jp1su
Apr 23 09:25:00 vrouter vrlld[3368]: healthcheck: Concurrent License Activated (3)

```

(continues on next page)

(continued from previous page)

```
Apr 23 09:25:00 vrouter vrld[3368]: computer id: mLgk6tN3nKRB5Z9Jp1su
vrouter running license#
```

In the nominal case, license key activation success is indicated by the following messages:

```
Apr 22 21:24:55 localhost vrld[3368]: healthcheck: Concurrent License Activated (3)
Apr 22 21:24:55 localhost vrld[3368]: computer id: mLgk6tN3nKRB5Z9Jp1su
Apr 22 21:24:55 localhost vrld[3368]: started
```

Then, health check occurs and succeeds regularly:

```
Apr 23 03:25:00 vrouter vrld[3368]: healthcheck: Concurrent License Activated (3)
Apr 23 03:25:00 vrouter vrld[3368]: computer id: mLgk6tN3nKRB5Z9Jp1su
Apr 23 09:25:00 vrouter vrld[3368]: healthcheck: Concurrent License Activated (3)
Apr 23 09:25:00 vrouter vrld[3368]: computer id: mLgk6tN3nKRB5Z9Jp1su
```

## Troubleshooting license activation failures

The following messages indicates that the licensing server could not be reached.

---

**Note:** The following hosts must be reachable: nsa.nalpeiron.com, my.nalpeiron.com.

---

- Server address cannot be resolved

```
vrouter> show license
Expired license.
vrouter> show log service license
(...)
Apr 23 15:53:29 vrouter vrld[24700]: healthcheck: cannot get license: SOAP Host_
↳can't be resolved (-4303)
Apr 23 15:53:29 vrouter vrld[24700]: License status from cache -1
Apr 23 15:53:29 vrouter vrld[24700]: license code: xxxxxxxxxxxxxxxx
Apr 23 15:53:29 vrouter vrld[24700]: healthcheck: Expired license (-1)
Apr 23 15:53:29 vrouter vrld[24700]: computer id: mLgk6tN3nKRB5Z9Jp1su
Apr 23 15:53:29 vrouter vrld[24700]: Failed to get maintenance expiry date:_
↳Lookup of tag failed (-6003)
(...)
```

- Server is not accessible

```
vrouter> show license
Expired license.
```

(continues on next page)

(continued from previous page)

```

vrouters> show log service license
(...)
Apr 27 13:55:14 vrouter vrld[4535]: healthcheck: cannot get license: SOAP Couldn
↳'t connect to host (-4304)
Apr 27 13:55:14 vrouter vrld[4535]: License status from cache -1
Apr 27 13:55:14 vrouter vrld[4535]: license code: xxxxxxxxxxxxxxxx
Apr 27 13:55:14 vrouter vrld[4535]: healthcheck: Expired license (-1)
Apr 27 13:55:14 vrouter vrld[4535]: computer id: mLgk6tN3nKRB5Z9Jp1su
Apr 27 13:55:14 vrouter vrld[4535]: Failed to get maintenance expiry date: Lookup
↳of tag failed (-6003)
(...)

```

The following messages indicate license key activation errors.

- Non-existing license key (likely wrong serial)

```

vrouters> show license
Expired license.
vrouters> show log service license
(...)
Apr 25 04:35:47 vrouter vrld[13156]: healthcheck: cannot get license: SOAP
↳invalid license number (-4212)
Apr 25 04:35:47 vrouter vrld[13156]: License status from cache -1
Apr 25 04:35:47 vrouter vrld[13156]: license code: xxxxxxxxxxxxxxxx
Apr 25 04:35:47 vrouter vrld[13156]: healthcheck: Expired license (-1)
Apr 25 04:35:47 vrouter vrld[13156]: computer id: 7A6YS9Kc+dfN3MGmuWJD
Apr 25 04:35:47 vrouter vrld[13156]: Failed to get maintenance expiry date:
↳Lookup of tag failed (-6003)
(...)

```

- Disabled license key

```

vrouters> show license
LicenseCode is inactive.
vrouters> show log service license
(...)
Apr 25 04:36:20 vrouter vrld[13852]: license code:
↳xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Apr 25 04:36:20 vrouter vrld[13852]: healthcheck: LicenseCode is inactive (-114)
Apr 25 04:36:20 vrouter vrld[13852]: computer id: d23nbSAwfJRVVWIBEGKo
Apr 25 04:36:20 vrouter vrld[13852]: Failed to get maintenance expiry date:
↳Lookup of tag failed (-6003)
(...)

```

- Maintenance expired prior to Turbo Router release date

```

vrouters> show license
Maintenance Expired.
vrouters> show log service license
(...)
Apr 25 04:36:53 vrouter vrld[14569]: Maintenance expiry date too old for version.
↳3.0.0.ga (xxxxxxxxxxxx)
Apr 25 04:36:53 vrouter vrld[14569]: Returning license to server
Apr 25 04:36:53 vrouter vrld[14569]: license code:
↳xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Apr 25 04:36:53 vrouter vrld[14569]: healthcheck: Maintenance Expired (-6001)
Apr 25 04:36:53 vrouter vrld[14569]: computer id: DL/g0a09evIUvk+ror1t
(...)

```

- Maximum number of activations reached

```

vrouters> show license
Number of Allowed Activations Exceeded.
vrouters> show log service license
(...)
Apr 25 04:37:26 vrouter vrld[15250]: license code:
↳xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Apr 25 04:37:26 vrouter vrld[15250]: healthcheck: Number of Allowed Activations.
↳Exceeded (-115)
Apr 25 04:37:26 vrouter vrld[15250]: computer id: ie128hMf9mXLC/P/hUtf
Apr 25 04:37:26 vrouter vrld[15250]: Failed to get maintenance expiry date:
↳Lookup of tag failed (-6003)
(...)

```

## License key cache

When the license key is replaced by a new one and activation fails, the previous license key remains valid until it expires. This is to prevent interruption of service in case of configuration mistake or temporary connection issue. After the new license key is successfully activated, the previous one is discarded.

However, activation may succeed but the resulting license key be invalid, for example if the key has been disabled on the server or has no activations left. In that case, the license will be discarded and the features and capacities will be restricted. It may be necessary to perform Day-1 configuration again to setup a valid license.

## Licensing usage report

Licensing usage report is automatically provided to the licensing server upon license activation and health check. It is also included in the troubleshooting report.

The license usage report includes the following information:

- average network throughput (measured in Rx)
- average and current number of IPsec tunnels
- average and current number of CG-NAT connections

## Licensing End-User Portal

The Licensing End-User Portal is accessible through the 6WIND Customer Zone. Your credentials should have been provided by the Customer Support Team when purchasing the Turbo Router software.

The Licensing End-User Portal allows to:

- review your license keys,
- check the number of instances activated for a license key,
- check the number of remaining activations for a license key,
- delete an activated instance (in case of crash for example),
- activate and deactivate a license offline,
- manage your licensing user portal accounts.

## Users

### Overview

Two user roles are available:

- **viewer** for use in operational mode where the configuration cannot be changed, only commands to troubleshoot or monitor are available.

This is the default role for new users.

- **admin** for use in configuration mode, with full access.

Three user accounts are provided by default:

| Account | Default password | Description  |
|---------|------------------|--|
| admin   | admin            | The standard account for configuration. It has the <code>admin</code> role.  |
| viewer  | viewer           | A restricted account for monitoring purposes. It has the <code>viewer</code> role.   |
| root    | 6windos          | Provides the ability to log into the Linux subsystem as superuser. Note that any configuration or customization of the vRouter in this mode is out of the support scope, may break the system, and will be lost after an update. |

**Warning:** For security reasons, it is recommended to change the default passwords of preconfigured users. See *Changing Passwords*.

Two default users are created when booting the system for the first time: `admin` and `viewer`. Their default passwords are `admin` and `viewer`, respectively.

The `admin` account has the `admin` role, which means that it has permissions to edit the configuration and run privileged commands.

The `viewer` account has the `viewer` role, which means that it has permissions to view the configuration but not to edit it and run standard commands.

**Warning:** For obvious security reasons, you **MUST** change the passwords of these users.

You may even want to completely disable the default `admin` and `viewer` users, by setting `default-users-enabled` to `false`:

```
vrouter running config# system auth default-users-enabled false
vrouter running config# commit
Configuration applied.
```

In this case, you must configure a user with the `admin` role, else you will lose access to the CLI.

## Changing Passwords

### CLI users

To change the `admin` user password, go in the `system auth user admin` context:

```
vrouter running config# system auth user admin
vrouter running user admin# password
Enter value for password> *****
vrouter running user admin# commit
Configuration applied.
```

For security reasons, the password is not stored in clear-text in the configuration. A hash is stored instead.

```
vrouter running user admin# show config
user admin
  password $5$Ndx/Q1MS5Anp7LTq$Lws20mAm0SO.cBmPBGtdpwnfdAM4hDM4AdS04ncXjS/
```

It is also possible to directly set the password as a hashed value. To generate a hashed password on a Linux machine, use `mkpasswd`, which is provided in the `whois` package:

```
root@host:~# mkpasswd -m SHA-256
Password: *****
$5$Ndx/Q1MS5Anp7LTq$Lws20mAm0SO.cBmPBGtdpwnfdAM4hDM4AdS04ncXjS/
```

## root user

Changing the password for the root user is done through the Linux shell:

```
root@vrouter:~# passwd
Enter new UNIX password: *****
Retype new UNIX password: *****
passwd: password updated successfully
```

## Creating Users

To create a new user, go into the `config system auth` context, and add a new user with the following commands:

```
vrouter running user admin# ..
vrouter running auth# user john
vrouter running user john# role admin
vrouter running user john# password
Enter value for password> *****
vrouter running user john# commit
Configuration applied.
```

Let's display what has been sent to the NETCONF server:

```
vrouter running user john# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <auth xmlns="urn:6wind:vrouter/system/auth">
      <user>
        <name>john</name>
        <role>admin</role>
```

(continues on next page)

(continued from previous page)

```

    <password>$5$iqsVCbCmIYRF.Sht$lCwP.HDLxtTnzz33uXX7ZdTR6xdSdnUoabRMxHYXjI9</
↵password>
    </user>
  </auth>
</system>
</config>

```

Now that the configuration is applied, let's see the state of our user:

```

vrouter running user john# show state
user john
  password $5$iqsVCbCmIYRF.Sht$lCwP.HDLxtTnzz33uXX7ZdTR6xdSdnUoabRMxHYXjI9
  role admin
  ..

```

The user `john` has the `admin` role. This means he can edit the configuration, read protected nodes (such as passwords) and run privileged commands.

## Configuring SSH Authorized Keys

SSH authentication can be used to login to Turbo Router without a password.

This requires to configure one or more `authorized-key`.

### Generating a key pair

First, you need to generate a key pair on a remote machine.

```

user@my-laptop:~$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase): *****
Enter same passphrase again: *****
Your identification has been saved in /home/user/.ssh/id_ecdsa.
Your public key has been saved in /home/user/.ssh/id_ecdsa.pub.
The key fingerprint is:
SHA256:UrMHdqPxmoE8V8DNYRtL0I15cWAFfzZn7PHy4j2enH5A robobuild@ubuntu1604es
The key's randomart image is:
+---[ECDSA 256]---+
|      .o+++..oo|
|      +o+ . oo|
|      0 0 o  .|
|      + ^ +  ..|

```

(continues on next page)

(continued from previous page)

```
|      . S O E.o. |
|      . * o oo+ |
|      + .   oo |
|      .   ..= |
|      o*+ |
+-----[SHA256]-----+
```

## Configuring an authorized-key for CLI users

Copy the public key file contents into the configuration:

```
user@my-laptop:~$ cat ~/.ssh/id_ecdsa.pub
ecdsa-sha2-nistp256_
↪AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBH2hK42JHtTYU1XRw2Zu4xCriM7CIXBl19p1/
↪1qkapobkS6yCnwauqTEveBw1G0jwuTADvqQVozBoaLbY3KGmsI= user@my-laptop
```

```
vrouter running user john# authorized-key "ecdsa-sha2-nistp256_
↪AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBH2hK42JHtTYU1XRw2Zu4xCriM7CIXBl19p1/
↪1qkapobkS6yCnwauqTEveBw1G0jwuTADvqQVozBoaLbY3KGmsI= user@my-laptop"
vrouter running user john# commit
Configuration applied.
```

**Warning:** NEVER copy the private key contents. Only the **PUBLIC** key.

## Configuring an authorized-key for root user

This is done from the Linux shell. Copy the public key file contents into the `/root/.ssh/authorized_keys` file:

```
user@my-laptop:~$ cat ~/.ssh/id_ecdsa.pub
ecdsa-sha2-nistp256_
↪AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBH2hK42JHtTYU1XRw2Zu4xCriM7CIXBl19p1/
↪1qkapobkS6yCnwauqTEveBw1G0jwuTADvqQVozBoaLbY3KGmsI= user@my-laptop

root@vrouter:~# cat >> /root/.ssh/authorized_keys
ecdsa-sha2-nistp256_
↪AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBH2hK42JHtTYU1XRw2Zu4xCriM7CIXBl19p1/
↪1qkapobkS6yCnwauqTEveBw1G0jwuTADvqQVozBoaLbY3KGmsI= user@my-laptop
<Ctrl+D>
root@vrouter:~# chmod 600 /root/.ssh/authorized_keys
```

## Checking the connection

After which you may check that the remote authentication works without a password:

```
user@my-laptop:~$ ssh -i ~/.ssh/id_ecdsa john@vrouter
The authenticity of host 'vrouter (10.0.0.58)' can't be established.
ECDSA key fingerprint is SHA256:nNerPB16BKwHmcex5IVKS7YMVt4VuaVavH3LId7uI6Q.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'vrouter,10.0.0.58' (ECDSA) to the list of known hosts.
Enter passphrase for key '/home/user/.ssh/id_ecdsa': *****
Welcome to Turbo Router - 3.2

vrouter>
```

---

**Note:** If you did set a passphrase on your private key, you will need to enter it.

---

### See also:

The *command reference* for details.

## Authentication, Authorization and Accounting (AAA)

### Overview

Users authentication can be done using a TACACS+ remote server.

Each remote user is assigned a role (*viewer* or *admin*, see *users* section for details) that denotes its rights. The way to specify this role is dependent of the remote server.

---

**Note:** If a local user with the same name as a remote user exists, the connection can be done by using the local or remote password. The role of the user will be the one defined locally.

---

**Warning:** Some names are reserved by the system and cannot be used: *\_apt*, *\_lldpd*, *\_tacacs*, *backup*, *bin*, *daemon*, *dhcpd*, *dnsmasq*, *fastpath*, *games*, *gnats*, *irc*, *list*, *lp*, *mail*, *man*, *messagebus*, *news*, *nobody*, *ntp*, *proxy*, *snmp*, *sshd*, *statd*, *sync*, *sys*, *syslog*, *systemd-bus-proxy*, *systemd-network*, *systemd-resolve*, *systemd-timesync*, *telegraf*, *uucp*, *uuidd*, *www-data*.

If one of these names is used, the connection using a remote server will fail.

## Manage TACACS+ servers list

To add a TACACS+ servers do:

```
vrouter running config# system aaa tacacs 1
```

Here, 1 is the priority order in case multiple servers are configured. The lower the order, the higher the priority.

---

**Note:** Up to 8 TACACS+ servers can be specified.

---

An IP address and secret to authenticate the TACACS+ exchanges are required:

```
vrouter running tacacs 1#! address 192.168.0.1 secret testing123
vrouter running tacacs 1# commit
```

**Warning:** The specified address must be accessible from vrf 'main'.

Let's fetch the state after committing this configuration:

```
vrouter running tacacs 1# show state
tacacs 1
  address 192.168.0.1
  port 49
  secret testing123
  timeout 3
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running config# show config xml absolute system aaa tacacs
<config xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <aaa xmlns="urn:6wind:vrouter/system/aaa">
      <tacacs>
        <order>1</order>
        <port>49</port>
        <timeout>3</timeout>
        <address>192.168.0.1</address>
        <secret>testing123</secret>
      </tacacs>
    </aaa>
  </system>
</config>
```

**See also:**

The *command reference* for details.

**Configuring TACACS+ authentication servers**

6WIND Vendor-Specific TACACS+ Attributes can be used to configure users privileges. They are specified in the TACACS+ server configuration file on a per-user basis. Turbo Router retrieves these attributes through an authorization request to the TACACS+ server after authenticating a user.

To specify these attributes, include a service statement in the TACACS+ server configuration file, in a user or a group statement:

```
service = 6WIND {  
    local-role = "admin|viewer"  
}
```

At the moment, the `local-role` attribute is supported. If not specified, the `viewer` role is assigned by default.

Here is a complete example:

```
group = admins {  
    default service = permit  
    service = exec {  
        priv-lvl = 15  
    }  
    service = shell {  
        priv-lvl = 15  
    }  
    service = 6WIND {  
        local-role = "admin"  
    }  
}  
  
group = viewers {  
    default service = permit  
    service = exec {  
        priv-lvl = 15  
    }  
    service = shell {  
        priv-lvl = 15  
    }  
    service = 6WIND {  
        local-role = "viewer"  
    }  
}
```

(continues on next page)

(continued from previous page)

```
user = john {
    name = "John C"
    member = admins
    pap = PAM
}

user = alice {
    default service = permit
    service = exec {
        priv-lvl = 15
    }
    service = shell {
        priv-lvl = 15
    }
    service = 6WIND {
        local-role = "admin"
    }
    name = "Alice F"
    pap = PAM
}

user = bob {
    name = "Bob D"
    member = viewers
    pap = PAM
}
```

With this configuration, john and alice can connect to the product with the admin role and bob with the viewer role.

---

**Note:** The length of the user name must be less or equal to 32 characters.

---

## Hostname

The device hostname can be changed.

To set the hostname to myhostname, do:

```
vrouter running config# system
vrouter running system# hostname myhostname
vrouter running system# commit
```

To display the hostname state:

```
myhostname> show state / system
system
  hostname myhostname
  (...)
```

The same configuration can be made using this NETCONF XML configuration:

```
myhostname running system# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <hostname>myhostname</hostname>
  </system>
</config>
```

## Timezone

The device timezone can be changed.

To set the timezone to Europe/Paris, do:

```
vrouter running config# system
vrouter running system# timezone Europe/Paris
vrouter running system# commit
```

To display the timezone state, and the date:

```
vrouter> show state / system
system
  timezone Europe/Paris
  date "Fri Jul 13 10:53:46 2018"
  (...)
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running system# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <timezone>Europe/Paris</timezone>
  </system>
</config>
```

## Network stack parameters

### IP/IPv6 parameters

The behavior of the IPv4/IPv6 network stack can be customized globally, and, for some parameters, per VRF. This behavior customization includes for instance the activation of forwarding, the filtering of packets with source routing option, etc...

If there is no configuration value in a VRF, the global configuration applies.

### Global configuration

To change the global default parameters, do:

```
vrouter running config# system network-stack ipv4
vrouter running ipv4# accept-redirects true
vrouter running ipv4# accept-source-route true
vrouter running ipv4# .. ipv6
vrouter running ipv6# accept-redirects true
vrouter running ipv6# accept-source-route true
vrouter running ipv6# accept-router-advert always
vrouter running ipv6# use-temporary-addresses always
vrouter running ipv6# commit
```

To display the global network stack parameters state:

```
vrouter> show state / system network-stack
network-stack
  icmp
    ignore-icmp-echo-broadcast false
    rate-limit-icmp 1000
    rate-mask-icmp destination-unreachable source-quench time-exceeded parameter-
    ↪problem
    ..
  ipv4
    forwarding true
    send-redirects true
    accept-redirects false
    accept-source-route false
    log-invalid-addresses false
    ..
  ipv6
    forwarding true
    accept-router-advert never
    use-temporary-addresses never
```

(continues on next page)

(continued from previous page)

```

accept-redirects false
accept-source-route false
..
..

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouters running network-stack# show config xml absolute
<config xmlns="urn:6wind:vrouters">
  <system xmlns="urn:6wind:vrouters/system">
    <network-stack>
      <ipv4>
        <forwarding>true</forwarding>
        <send-redirects>true</send-redirects>
        <accept-redirects>true</accept-redirects>
        <accept-source-route>true</accept-source-route>
        <log-invalid-addresses>false</log-invalid-addresses>
      </ipv4>
      <icmp>
        <ignore-icmp-echo-broadcast>false</ignore-icmp-echo-broadcast>
        <rate-limit-icmp>1000</rate-limit-icmp>
        <rate-mask-icmp>destination-unreachable source-quench time-exceeded parameter-
        ↪problem</rate-mask-icmp>
      </icmp>
      <ipv6>
        <forwarding>true</forwarding>
        <accept-router-advert>always</accept-router-advert>
        <use-temporary-addresses>always</use-temporary-addresses>
        <accept-redirects>true</accept-redirects>
        <accept-source-route>true</accept-source-route>
      </ipv6>
    </network-stack>
  </system>
</config>

```

## VRF configuration

To override the parameters for a specific VRF, do:

```

vrouters running config# vrf vr1 network-stack ipv4
vrouters running ipv4# accept-redirects false
vrouters running ipv4# .. ipv6
vrouters running ipv6# accept-redirects false

```

(continues on next page)

(continued from previous page)

```
vrouter running ipv6# commit
```

To display the network stack parameters state for this VRF:

```
vrouter running ipv6# show state / vrf vr1 network-stack
network-stack
  icmp
    ignore-icmp-echo-broadcast false
    rate-limit-icmp 1000
    rate-mask-icmp destination-unreachable source-quench time-exceeded parameter-
    ↪problem
    ..
  ipv4
    forwarding true
    send-redirects true
    accept-redirects false
    accept-source-route false
    log-invalid-addresses false
    ..
  ipv6
    forwarding true
    accept-router-advert never
    use-temporary-addresses never
    accept-redirects false
    accept-source-route false
    ..
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running network-stack# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>vr1</name>
    <network-stack xmlns="urn:6wind:vrouter/system">
      <icmp/>
      <ipv4>
        <accept-redirects>false</accept-redirects>
      </ipv4>
      <ipv6>
        <accept-redirects>false</accept-redirects>
        <accept-router-advert>never</accept-router-advert>
        <use-temporary-addresses>never</use-temporary-addresses>
      </ipv6>
    </network-stack>
  </vrf>
</config>
```

(continues on next page)

(continued from previous page)

```

</network-stack>
</vrf>
</config>

```

## Neighbor

The maximum number of neighbors entries is limited.

To change these limits, do:

```

vrouters running config# system
vrouters running system# network-stack
vrouters running network-stack# neighbor
vrouters running neighbor# ipv4-max-entries 4096
vrouters running neighbor# ipv6-max-entries 4096
vrouters running neighbor# commit

```

**Warning:** If the fast path is running, a similar change is required in *fast path limits configuration*.

To display the neighbor state:

```

vrouters> show state / system network-stack neighbor
neighbor
  ipv4-max-entries 1024
  ipv6-max-entries 1024
  ..

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouters running neighbor# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <network-stack>
      <neighbor>
        <ipv4-max-entries>4096</ipv4-max-entries>
        <ipv6-max-entries>4096</ipv6-max-entries>
      </neighbor>
    </network-stack>
  </system>
</config>

```

## Connection Tracking

The maximum number of connection tracking objects (used for IP filtering) is limited.

To change this limit, do:

```
vrouter running config# system
vrouter running system# network-stack
vrouter running network-stack# contrack
vrouter running contrack# max-entries 1000000
vrouter running contrack# commit
```

**Warning:** If the fast path is running, a similar change is required in *fast path limits configuration*.

To customize contrack TCP/UDP timeouts:

```
vrouter running config# system
vrouter running system# network-stack
vrouter running network-stack# contrack
vrouter running contrack# tcp-timeout-close 20
vrouter running contrack# tcp-timeout-close-wait 70
vrouter running contrack# tcp-timeout-established 500000
vrouter running contrack# tcp-timeout-fin-wait 130
vrouter running contrack# tcp-timeout-last-ack 40
vrouter running contrack# tcp-timeout-max-retrans 400
vrouter running contrack# tcp-timeout-syn-recv 70
vrouter running contrack# tcp-timeout-syn-sent 130
vrouter running contrack# tcp-timeout-time-wait 130
vrouter running contrack# tcp-timeout-unacknowledged 400
vrouter running contrack# udp-timeout 40
vrouter running contrack# udp-timeout-stream 190
vrouter running contrack# commit
```

To display the contrack state:

```
vrouter> show state / system network-stack contrack
contrack
  max-entries 1000000
  tcp-timeout-close 20
  tcp-timeout-close-wait 70
  tcp-timeout-established 500000
  tcp-timeout-fin-wait 130
  tcp-timeout-last-ack 40
  tcp-timeout-max-retrans 400
```

(continues on next page)

(continued from previous page)

```

tcp-timeout-syn-recv 70
tcp-timeout-syn-sent 130
tcp-timeout-time-wait 130
tcp-timeout-unacknowledged 400
udp-timeout 40
udp-timeout-stream 190
..

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouters running contrack# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <network-stack>
      <contrack>
        <max-entries>10000000</max-entries>
        <tcp-timeout-close>20</tcp-timeout-close>
        <tcp-timeout-close-wait>70</tcp-timeout-close-wait>
        <tcp-timeout-fin-wait>130</tcp-timeout-fin-wait>
        <tcp-timeout-last-ack>40</tcp-timeout-last-ack>
        <tcp-timeout-max-retrans>400</tcp-timeout-max-retrans>
        <tcp-timeout-syn-recv>70</tcp-timeout-syn-recv>
        <tcp-timeout-syn-sent>130</tcp-timeout-syn-sent>
        <tcp-timeout-time-wait>130</tcp-timeout-time-wait>
        <tcp-timeout-unacknowledged>400</tcp-timeout-unacknowledged>
        <udp-timeout>40</udp-timeout>
        <udp-timeout-stream>190</udp-timeout-stream>
      </contrack>
    </network-stack>
  </system>
</config>

```

## Fast path

The fast path is the Turbo Router component in charge of packet processing acceleration. There is only one instance of fast path, that can manage interfaces in several VRF.

## Enable the fast path

To accelerate ethernet NICs, they must be dedicated to the fast path, and the fast path must be started:

```
vrouters> edit running
vrouters running config# system fast-path
vrouters running fast-path#! port pci-b0s4
vrouters running fast-path# port pci-b0s5
vrouters running fast-path# show config
fast-path
  enabled true
  port pci-b0s4
  port pci-b0s5
  cp-protection
    budget 10
  ..
vrouters running fast-path# commit
```

---

**Note:** use `show state / network-port` to see the list of available network ports with PCI ids; it can help choosing the right ports.

---

The same configuration can be made using this NETCONF XML configuration:

```
vrouters running config# show config xml absolute system fast-path
<config xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <fast-path xmlns="urn:6wind:vrouter/fast-path">
      <enabled>true</enabled>
      <cp-protection>
        <budget>10</budget>
      </cp-protection>
      <port>pci-b0s4</port>
      <port>pci-b0s5</port>
      <core-mask/>
      <crypto/>
      <advanced/>
      <limits/>
    </fast-path>
  </system>
</config>
```

Check the current state of the fast path:

```
vrouter running fast-path# show state
fast-path
port pci-b0s5
port pci-b0s4
enabled true
core-mask
  fast-path 2-3
  exception 0
  linux-to-fp 2-3
  ..
cpu-usage cpu2
  busy 0
  ..
cpu-usage cpu3
  busy 0
  ..
cp-protection
  budget 10
  ..
crypto
  nb-session 0
  nb-buffer 0
  ..
advanced
  nb-mbuf 32768
  offload false
  vlan-strip false
  intercore-ring-size 128
  software-txq 0
  reserve-hugepages true
  ipv4-netfilter-cache true
  ipv6-netfilter-cache true
  ipv4-pre-ipsec-fragmentation off
  ipv6-pre-ipsec-fragmentation off
  ..
limits
  fp-max-vrf 16
  ..
```

---

**Note:** fast path starting can take several seconds.

---

## Configuring the core masks

In the `core-mask` context, the assignment of cores can be customized. This includes:

- The cores which are dedicated to the fast path for dataplane operations. The accepted values are either a policy (`min`, `half`, `max`) or a core mask. By default, half of the available cores on are dedicated to the fast path for dataplane operations.
- Which dataplane cores (included in fast path mask) that receive packets from Linux. By default, all dataplane cores.
- The control plane cores (disjoint of fast path mask) that receive exception packets. By default, the first control plane core.
- The mapping between fast path cores and the ports, in other words which core polls which port. By default, each port is polled by each core of the same NUMA node.

Here is an example of configuration with a custom fast path core mask and exception mask:

```
vrouter> edit running
vrouter running config# system fast-path
vrouter running fast-path#! port pci-b0s4
vrouter running fast-path# core-mask
vrouter running core-mask# fast-path 5,9-12
vrouter running core-mask# exception 0-4
vrouter running core-mask# ..
vrouter running fast-path# show config
fast-path
  enabled true
  port pci-b0s4
  core-mask
    fast-path 5,9-12
    exception 0-4
    ..
  cp-protection
    budget 10
    ..
  ..
vrouter running fast-path# commit
```

---

**Note:** use `show state / system linux` to see the list of available cores.

---

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running config# show config xml absolute system fast-path
<config xmlns="urn:6wind:vrouter">
```

(continues on next page)

(continued from previous page)

```

<system xmlns="urn:6wind:vrouter/system">
  <fast-path xmlns="urn:6wind:vrouter/fast-path">
    <enabled>true</enabled>
    <core-mask>
      <fast-path>5,9-12</fast-path>
      <exception>0-4</exception>
    </core-mask>
    <cp-protection>
      <budget>10</budget>
    </cp-protection>
    <crypto/>
    <advanced/>
    <limits/>
    <port>pci-b0s4</port>
  </fast-path>
</system>
</config>

```

### Fast path limits configuration

The fast path capabilities can be tuned according to your requirements in terms of scalability and memory footprint. This is done through the fast path limits configuration.

Here is an example of configuration with a custom number of VRs and IPv4 routes:

```

vrouters> edit running
vrouters running config# system fast-path
vrouters running fast-path#! port pci-b0s4
vrouters running fast-path# limits
vrouters running limits# fp-max-vrf 128
vrouters running limits# ip4-max-route 1000000
vrouters running limits# ..
vrouters running fast-path# show config
fast-path
  enabled true
  port pci-b0s4
  cp-protection
    budget 10
  ..
limits
  fp-max-vrf 128
  ip4-max-route 1000000
  ..

```

(continues on next page)

(continued from previous page)

```
..
vrouter running fast-path# commit
```

**Warning:** Similar changes may be required in *system neighbor configuration* and in *system conntrack configuration*.

**Note:** Default fast path scalability limits are automatically adjusted if memory is insufficient, to prevent startup failure due to lack of memory. `show state / system fast-path limits` can be used to check the actual values.

The same configuration can be made using this NETCONF XML configuration:

```
dut-vm running config# show config xml absolute system fast-path
<config xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <fast-path xmlns="urn:6wind:vrouter/fast-path">
      <enabled>true</enabled>
      <core-mask/>
      <cp-protection>
        <budget>10</budget>
      </cp-protection>
      <crypto/>
      <advanced/>
      <limits>
        <fp-max-vrf>128</fp-max-vrf>
        <ip4-max-route>1000000</ip4-max-route>
      </limits>
      <port>pci-b0s4</port>
    </fast-path>
  </system>
</config>
```

## Advanced fast path configuration

For advanced users, some fast path parameters can also be customized: the number of network packet buffers, the number of crypto buffers or sessions, the activation of advanced offload features, the exception core mask, the hardware queue mapping etc. . .

Please refer to the fast path *crypto command reference* and the fast path *advanced command reference* for details.

## Control Plane Protection

In a network architecture, control packets are critical, since losing some of them has stronger consequences than losing data packets:

- losing ARP (Address Resolution Protocol) packets can make a gateway unreachable
- losing OSPF/BGP/... packets can make a network unreachable
- losing IKE packets can prevent the setup of IPSEC security associations

*Control Plane Protection* is a software mechanism that reduces the risk of dropping these control packets. It has an impact on performance, which can be tuned depending on the required throughput and criticality of losing control packets.

The software parser recognizes ARP, ICMP (Internet Control Message Protocol), ICMPv6, OSPF, VRRP, IKE, DHCP, DHCPv6, BGP, LACP (Link Aggregation Control Protocol), SSH, OpenFlow, JSON RPC (TCP (Transmission Control Protocol) port 7406), Stats Collector (TCP port 39090), DPVI (Data Plane Virtual Interface) packets. All can be encapsulated in VLAN, QinQ or FPTUN (Fast Path Tunneling Protocol).

*Control Plane Protection* is disabled by default. It can be enabled on a per-interface basis, for RX (Reception) or TX (Transmission), depending on the situation:

- RX: the router is overloaded, the software is not able to dequeue the incoming packets fast enough, the hardware RX ring becomes full and the NIC starts to drop packets.
- TX: the router tries to send more packets than what the network link supports, the hardware TX ring becomes full and the software starts to drop packets.

*Control Plane Protection* works according to a maximum CPU budget. If control plane packets are still dropped after enabling *Control Plane Protection*, it means that this budget has to be increased.

To enable *Control Plane Protection* on a physical interface:

```
vrouter running config# system fast-path
vrouter running fast-path#! port pci-b0s4
vrouter running fast-path# cp-protection budget 10
vrouter running fast-path# / vrf main interface physical eth0
vrouter running physical eth0#! port pci-b0s4
vrouter running physical eth0# rx-cp-protection true
vrouter running physical eth0# tx-cp-protection true
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running config# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <fast-path xmlns="urn:6wind:vrouter/fast-path">
      <enabled>true</enabled>
      <core-mask/>
```

(continues on next page)

(continued from previous page)

```

    <cp-protection>
      <budget>10</budget>
    </cp-protection>
  <crypto/>
  <advanced/>
  <limits/>
  <port>pci-b0s4</port>
</fast-path>
</system>
<vrf>
  <name>main</name>
  <interface xmlns="urn:6wind:vrouter/interface">
    <physical>
      <name>eth0</name>
      <enabled>true</enabled>
      <ipv4>
        <enabled>true</enabled>
      </ipv4>
      <ipv6>
        <enabled>true</enabled>
      </ipv6>
      <ethernet>
        <auto-negotiate>true</auto-negotiate>
      </ethernet>
      <port>pci-b0s4</port>
      <rx-cp-protection>true</rx-cp-protection>
      <tx-cp-protection>true</tx-cp-protection>
    </physical>
  </interface>
</vrf>
</config>

```

---

**Note:** the *Control Plane Protection* feature only works when the fast path is enabled, if the feature is supported by the NIC driver.

---

*Control Plane Protection* provides statistics to monitor the number of filtered packets:

```

vrouter running fast-path# show interface hardware-statistics eth0
(...)
fpn.rx_cp_passthrough: 0
fpn.rx_cp_kept: 0
fpn.rx_dp_drop: 0

```

(continues on next page)

(continued from previous page)

```

fpn.rx_cp_overrun: 0
fpn.tx_cp_passthrough: 0
fpn.tx_cp_kept: 0
fpn.tx_dp_drop: 0
fpn.tx_cp_overrun: 0
(...)

```

When *RX Control Plane Protection* is enabled, `fpn.rx_cp_passthrough` is increased for each received packet when machine is not overloaded. These packets are processed normally without being analyzed.

If the machine is loaded (RX ring length exceeds the threshold) and the CPU budget is not reached, `fpn.rx_cp_kept` and `fpn.tx_dp_drop` will increase respectively for each control plane packet (kept) and for each data plane packet (drop).

If the CPU budget is exceeded, `fpn.rx_cp_overrun` is increased for each received packet. These packets are processed normally without being analyzed.

The same applies for TX.

**See also:**

The *command reference* for details.

### Isolation of dataplane cores

The cores that are in charge of processing the network packets (the data plane) are dedicated to this task. The other tasks (the control plane) run on the other cores.

To display the cores affected to control plane:

```

vrouter> show state system cp-mask
cp-mask 0-2

```

To change the cores affected to control plane:

```

vrouter> edit running
vrouter running config# system cp-mask 0
vrouter running config# commit
Configuration committed.

```

---

**Note:** It is not possible to add fast path cores in cp-mask.

---



---

**Important:** To get the best performance, fast path cores should be isolated thanks to the cmd

---

`set-next-boot-params isolate-cpus <fp-coremask>` command. A reboot is needed after the `set-next-boot-params` command has been issued.

---

## SSH

Secure Shell (SSH) server can be configured for remote login to the router, giving a secure connection from standard SSH clients.

Independent SSH servers can be started in any vrf.

Users can configure the address and port on which SSH listens on.

Here is an example of configuration that will start a SSH server in the main vrf:

```
vrouter running config# vrf main
vrouter running vrf main# ssh-server
vrouter running ssh-server# commit
Configuration applied.
```

To display the SSH server state:

```
vrouter running config# show state vrf main ssh-server
ssh-server
  port 22
  enabled true
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running config# show config xml absolute vrf main ssh-server
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <ssh-server xmlns="urn:6wind:vrouter/ssh-server">
      <enabled>true</enabled>
      <port>22</port>
    </ssh-server>
  </vrf>
</config>
```

### See also:

The *command reference* for details.

## NETCONF server

As explained in *the introduction*, Turbo Router provides a NETCONF API that is used by NETCONF clients to configure and monitor the router remotely.

At startup, if the NETCONF server is not configured, it listens on all interfaces (IPv4 (Internet Protocol version 4) and IPv6 (Internet Protocol version 6) addresses) on port 830 of the main VRF.

The VRF, IP address and port on which the NETCONF server listens can be configured. This replaces the default configuration.

Here is an example of configuration that will start the NETCONF server in the `mgmt` VRF on addresses `192.168.0.5`, port `8030` and `fec0::dcad:cafe:ae01:203`, port `830`:

```
vrouter running config# vrf mgmt netconf-server
vrouter running netconf-server# enabled true
vrouter running netconf-server# address 192.168.0.5 port 8030
vrouter running netconf-server# address fec0::dcad:cafe:ae01:203
vrouter running netconf-server# commit
Configuration applied.
```

To display the NETCONF server state:

```
vrouter running config# show state vrf mgmt netconf-server
netconf-server
  enabled true
  address 192.168.0.5 port 8030
  address fec0::dcad:cafe:ae01:203 port 830
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running config# show config xml absolute vrf main netconf-server
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>mgmt</name>
    <netconf-server xmlns="urn:6wind:vrouter/netconf-server">
      <enabled>true</enabled>
      <address>
        <ip>192.168.0.5</ip>
        <port>8030</port>
      </address>
      <address>
        <ip>fec0::dcad:cafe:ae01:203</ip>
        <port>830</port>
      </address>
    </netconf-server>
  </vrf>
</config>
```

(continues on next page)

(continued from previous page)

```
</vrf>  
</config>
```

Here is an example of configuration that will delete the previous configuration and reestablish the default behavior (listen on all interfaces on port 830):

```
vrouter running config# del vrf mgmt netconf-server  
vrouter running config# vrf main netconf-server  
vrouter running netconf-server# enabled true  
vrouter running netconf-server# commit  
Configuration applied.
```

Here is an example of configuration that will stop the NETCONF server:

```
vrouter running config# vrf main netconf-server  
vrouter running netconf-server# enabled false  
vrouter running netconf-server# commit  
Configuration applied.
```

**Attention:** If you disable the NETCONF server, any remote operation via NETCONF will be made impossible. If you have not disabled it into the startup configuration, a reboot will restore the default configuration.

If you have explicitly *disabled* the NETCONF server in your startup configuration, remote NETCONF operation will *not* be enabled on boot. SSH remote access is not related to this and remains available unless you also disabled it.

### See also:

The *command reference* for details.

## NTP

Network Time Protocol (NTP (Network Time Protocol)) is a networking protocol for clock synchronization. Basically the required parameters are the peer(s) with which you accept to exchange information, and the frequency of updates.

Only one NTP client can be enabled at a time.

Here is an example on querying one NTP server with the parameter `iburst` set to enable burst synchronization:

```
vrouter running config# vrf main  
vrouter running vrf main# ntp  
vrouter running ntp# server my.timeserver.com iburst true  
vrouter running ntp# commit
```

To check the state:

```
vrouter running config# show state vrf main ntp
ntp
  server my.timeserver.com
    synchronized true
    stratum 6
    offset 19
    state system-peer
    version 4
    association-type SERVER
    root-delay 340
    iburst true
    prefer false
    root-dispersion 29
  ..
..
```

To show the state in a human readable way:

```
vrouter running config# show ntp vrf main
NTP synchronized with my.timeserver.com at stratum 6.
time correct within 19 ms.
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running config# show config xml absolute vrf main ntp
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <ntp xmlns="urn:6wind:vrouter/ntp">
      <enabled>true</enabled>
      <server>
        <address>my.timeserver.com</address>
        <iburst>true</iburst>
        <version>4</version>
        <association-type>SERVER</association-type>
        <prefer>false</prefer>
      </server>
    </ntp>
  </vrf>
</config>
```

**See also:**

The *command reference* for details.

## DNS client

Domain Name Service (DNS) provides name to IP address mapping.

Here is an example of DNS configuration to send DNS queries to the 192.168.0.254 server, and search for example.local domain.

```
vrouter running config# vrf main
vrouter running vrf main# dns
vrouter running dns# server 192.168.0.254
vrouter running dns# search example.local
vrouter running dns# commit
```

To display the DNS client state:

```
vrouter running config# show state vrf main dns
dns
  search example.local
  server address 192.168.0.254
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running config# show config xml absolute vrf main dns
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <dns xmlns="urn:6wind:vrouter/dns">
      <server>
        <address>192.168.0.254</address>
      </server>
      <search>example.local</search>
    </dns>
  </vrf>
</config>
```

### See also:

The *command reference* for details.

## Login banner

When logging in to the system from the console or via ssh, a pre-login banner is displayed before the login prompt, and a post-login banner is displayed after successfully logging in.

These banners may be customized.

## Set custom banners

To specify a custom pre-login banner, type the following command:

```
vrouter> cmd banner pre-login message "WARNING! ACCESS RESTRICTED"  
OK.  
vrouter>
```

To specify a custom post-login banner, type the following command:

```
vrouter> cmd banner post-login message "Welcome to the management network"  
OK.  
vrouter>
```

The effect of these commands is as follows:

```
vrouter> exit  
  
WARNING! ACCESS RESTRICTED  
vrouter login: admin  
Password:  
Last login: Tue Jul  9 12:57:10 UTC 2019 on ttyS0  
Welcome to the management network  
vrouter >
```

## Restore factory banners

To restore the factory pre-login banner, type the following command:

```
vrouter> cmd banner pre-login reset  
OK.  
vrouter>
```

To restore the factory post-login banner, type the following command:

```
vrouter> cmd banner post-login reset  
OK.  
vrouter>
```

**See also:**

The *command reference* for details.

**Rebooting**

To reboot the machine, run the following command in operational mode:

```
vrouter> cmd reboot
System will reboot on Tue 2020-03-17 14:07:15
vrouter>
```

Unless you specified otherwise, you have 60 seconds to cancel the reboot with the following command:

```
vrouter> cmd reboot cancel
Broadcast message from root@vrouter (Tue 2020-03-17 14:06:18 CET):

The system shutdown has been cancelled

Reboot cancelled.
vrouter>
```

A reboot is not possible if the *startup* configuration is not the same as the *running* configuration:

```
vrouter> cmd reboot
ERROR: Reboot cancelled: startup and running configurations are different.
Copy running to startup or use the "force" argument to bypass this check.
vrouter>
```

In this case, the *startup* configuration can be updated using `copy running startup`, or the use the `force` argument:

```
vrouter> cmd reboot
System will reboot on Tue 2020-03-17 14:07:15
vrouter>
```

**See also:**

The *command reference* for details.

## Powering Off

To completely shutdown the machine, run the following command in operational mode:

```
vrouter> cmd poweroff
System will poweroff on Tue 2020-03-17 14:07:15
vrouter>
```

Unless you specified otherwise, you have 60 seconds to cancel the operation with the following command:

```
vrouter> cmd poweroff cancel
Broadcast message from root@vrouter (Tue 2020-03-17 14:06:18 CET):

The system shutdown has been cancelled

Poweroff cancelled.
vrouter>
```

A poweroff is not possible if the *startup* configuration is not the same as the *running* configuration:

```
vrouter> cmd poweroff
ERROR: Poweroff cancelled: startup and running configurations are different.
Copy running to startup or use the "force" argument to bypass this check.
vrouter>
```

In this case, the *startup* configuration can be updated using `copy running startup`, or the use the `force` argument:.

```
vrouter> cmd poweroff
System will poweroff on Tue 2020-03-17 14:07:15
vrouter>
```

### See also:

The *command reference* for details.

## System Image

To manage system images, use the following commands in operational mode.

### List currently installed images

```
vrouter> cmd system-image list
3.0.0 (default) (current)
vrouter>
```

One image is displayed per line in the following format:

```
<image name> [(default)] [(current)] [(next)]
```

**<image name>** Name used to identify the image. If not set, the version is used.

**(default)** Set if it is the default boot image.

**(current)** Set if it is the image on which the system is booted.

**(next)** Set if the image will be used for the next boot.

### Download and install a new version

```
vrouter> cmd system-image import http://1.0.0.1:8000/6wind-vrouter-tr-ae-x86_64-
↪v3.0.1.update
  % Total      % Received % Xferd  Average Speed   Time    Time       Time  Current
             Dload  Upload   Total     Spent    Left     Speed
100 240M 100 240M    0      0  140M      0  0:00:01  0:00:01  --:--:-- 140M
vrouter> cmd system-image list
3.0.0 (default) (current)
3.0.1 (next)
vrouter>
```

---

**Note:** The newly installed image becomes the next boot image, but does not automatically become the default boot image.

This enables to test the installed image at next reboot. In case of problem, resetting the system will boot the default image.

Of course, you can explicitly set the newly installed image as the default image whenever you wish.

---

### Rename an image

```
vrouter> cmd system-image rename 3.0.1 new-name my-img
vrouter> cmd system-image list
3.0.0 (default) (current)
my-img (next)
vrouter>
```

### Change the default boot image

```
vrouter> cmd system-image set-default my-img
vrouter> cmd system-image list
3.0.0 (current)
my-img (default) (next)
vrouter>
```

### Remove an image

```
vrouter> cmd system-image delete my-img
vrouter> cmd system-image list
3.0.0 (default) (current)
vrouter>
```

---

**Note:** If the default boot image is deleted, the current image automatically becomes the default.

---

### See also:

The *command reference* for details.

## Logging

This section covers the configuration of the logging service. To display log messages, refer to the *show log documentation*.

### Local Logging Configuration

It is possible to configure the rate limiting that is applied to all messages generated on the system by changing rate limit interval and burst values.

```
vrouter running config# / system logging
vrouter running logging# rate-limit interval 20 burst 2000
vrouter running logging# commit
```

If, in the time interval defined by `interval` (in seconds), more messages than specified in `burst` are logged by a service, all further messages within the interval are dropped until the interval is over. A message about the number of dropped messages is generated. This rate limiting is applied per-service, so that two services which log do not interfere with each other's limits.

Defaults to 1000 messages in 30s.

To turn off any kind of rate limiting, set either value to 0.

Let's check the rate limit values have been applied properly:

```
vrouter running config# show state / system logging
logging
  rate-limit
    interval 20
    burst 2000
    ..
  disk-usage 6.1M
  ..
```

Note that `disk-usage` shows the sum of the file system usage of all archived and active journal files. The journal size is limited to half the size of the partition where the logs are located, up to a maximum of 4GB.

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running config# show config xml absolute / system logging
<config xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <logging xmlns="urn:6wind:vrouter/logging">
      <rate-limit>
        <interval>20</interval>
        <burst>2000</burst>
      </rate-limit>
    </logging>
  </system>
</config>
```

#### See also:

The *command reference* for details about the API, and the *show-log* command.

## Remote Syslog Configuration

syslog is a standard for message logging. It allows separation of the software that generates messages, the system that stores them, and the software that reports and analyzes them. Each message is labeled with a facility code, indicating the software type generating the message, and assigned a level.

Here we explain how to setup remote logging to a distant server.

## Client Configuration

The syslog client can be configured for sending log messages to remote servers:

```
vrouter running config# / vrf main logging syslog
vrouter running syslog#! remote-server 10.125.0.2 protocol tcp port 514
vrouter running syslog# commit
```

In this example, logs will be sent in TCP to remote server at address 10.125.0.2 and remote port 514 (which is the default).

To check the values have been applied in the system:

```
vrouter running config# show state / vrf main logging syslog
syslog
  enabled true
  remote-server 10.125.0.2
```

(continues on next page)

(continued from previous page)

```
protocol tcp
port 514
log-filter facility any
..
..
```

## Server Configuration

Here we provide an example configuration for the distant log server.

We assume the server is running Ubuntu 16.04 and that the rsyslog package is installed.

Open the rsyslog configuration file:

```
# vi /etc/rsyslog.conf
```

Find and uncomment the following lines to make your server to listen on the udp and tcp ports:

```
[...]
# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")
[...]
# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514")
[...]
```

Create a template file where we will create a new custom log format under the /etc/rsyslog.d/ directory:

```
# vi /etc/rsyslog.d/tmpl.conf
```

Add the following lines:

```
$template TmplAuth, "/var/log/client_logs/%HOSTNAME%/%PROGRAMNAME%.log"
$template TmplMsg, "/var/log/client_logs/%HOSTNAME%/%PROGRAMNAME%.log"

authpriv.* ?TmplAuth
*.info;mail.none;authpriv.none;cron.none ?TmplMsg
```

Reload the rsyslog service:

```
# systemctl restart rsyslog
```

**See also:**

The *command reference* for details about the API.

**Remote Log Filtering Configuration**

Logs sent to the remote servers can be configured using the `log-filter` command in the `remote-server` sub-context.

```
vrouter running remote-server 10.125.0.2# log-filter facility <NAME> level <LEVEL>
```

The first argument of the `log-filter` command is the facility on which the filter will be applied. Here is the list of available facilities:

| facility name | purpose   |
|---------------|---|
| any           | messages coming from ALL facilities                 |
| kernel        | messages coming from the kernel                     |
| mail          | messages coming from mail system                    |
| user          | user-level messages                                 |
| auth          | security/authorization messages                     |
| authpriv      | security/authorization messages (private)           |
| cron          | messages coming from the cron daemon                |
| daemon        | messages coming from daemons                        |
| FTP           | messages coming from the FTP daemon                 |
| syslog        | messages generated internally by the logging daemon |
| uucp          | messages coming from the Unix to Unix Copy Protocol |

The second argument is the log level. This argument can be a single syslog severity, a list of severities, or a severity preceded by `greater-or-equal`. `not` can also be used to negate a severity. `none` discards all messages for the facility. By default all messages are selected.

This table introduces the list of syslog severities from the most serious to the least:

| syslog severity | purpose                          |
|-----------------|----------------------------------|
| emergency       | system is unusable               |
| alert           | action must be taken immediately |
| critical        | critical conditions              |
| error           | error conditions                 |
| warning         | warning conditions               |
| notice          | normal but significant condition |
| info            | informational messages           |
| debug           | debug-level messages             |

**Note:**

- By default, if no filters are set, all log messages from all facilities are sent to the remote server. This implicit filter rule is replaced as soon as a rule is set.
- Each service has its own logging policy with regards to syslog facilities and severities. Refer to the services' documentation for details.

Here are some examples:

Send all messages greater or equal to error for all facilities:

```
vrouter running remote-server 10.125.0.2# log-filter facility any level greater-or-
↳equal error
```

Send all messages from the kernel facility:

```
vrouter running remote-server 10.125.0.2# log-filter facility kernel level any
```

Restrict the auth facility to emergency level:

```
vrouter running remote-server 10.125.0.2# log-filter facility auth level emergency
```

Disable all messages coming from the mail facility:

```
vrouter running remote-server 10.125.0.2# log-filter facility mail level none
```

## Transport Layer Security Configuration

The TLS (Transport Layer Security) configuration enables syslog messages encryption and servers authentication.

Entering the `tls` sub-context:

```
vrouter running config# / vrf main logging syslog tls
vrouter running tls#!
```

## Client Configuration

Configure the server authentication mode:

```
vrouter running tls# server-authentication
  anonymous|(name NAME [NAME [...]])|(fingerprint FP [FP [...]])|certificate
```

**anonymous** The servers are not authenticated.

**name NAME** The servers are authenticated if their certificate's common name match. Many names can be set.

**fingerprint FP** The servers are authenticated if their certificate's fingerprint match. Many fingerprints can be set.

**certificate** Validate only the server's certificate.

---

**Note:** Only one server authentication mode can be chosen at a time.

---

Add the certificate authority's certificate:

```
vrouter running tls# ca-certificate CERT
```

**CERT** The CA certificate between quotes.

The following options (**certificate** and **private-key**) are optional if the server doesn't authenticate its clients.

Add the client's certificate:

```
vrouter running tls# certificate CERT
```

**CERT** The client's certificate between quotes.

Add the client's private key:

```
vrouter running tls# private-key KEY
```

**KEY** The client's certificate key between quotes.

---

**Note:**

- The certificate and the private key have to be generated from the CA.
  - A minimal configuration is to add the **certificate-authority** and set the **server-authentication** option to **anonymous**, which is not recommended as servers and clients are not authenticated.
- 

## Server Configuration

See the rsyslog web documentation ([https://www.rsyslog.com/doc/v8-stable/tutorials/tls\\_cert\\_server.html](https://www.rsyslog.com/doc/v8-stable/tutorials/tls_cert_server.html)) for details about how to configure an rsyslog server with TLS encryption.

## Configuration Example

```
vrouter> edit running
vrouter running config# vrf main
vrouter running vrf main# logging syslog
vrouter running syslog#! remote-server 10.125.0.2
vrouter running remote-server 10.125.0.2# protocol tcp
```

(continues on next page)

(continued from previous page)

```

vrouters running remote-server 10.125.0.2# port 514
vrouters running remote-server 10.125.0.2# log-filter facility any level greater-or-
→equal error
vrouters running remote-server 10.125.0.2# log-filter facility kernel level any
vrouters running remote-server 10.125.0.2# log-filter facility auth level emergency
vrouters running remote-server 10.125.0.2# log-filter facility mail level none
vrouters running remote-server 10.125.0.2# ..
vrouters running syslog# tls
vrouters running tls#! ca-certificate "-----BEGIN CERTIFICATE-----
... MIID3zCCAkegAwIBAgIIXH6dxQIfVrcwDQYJKoZIhvcNAQELBQAwDTELMAkGA1UE
... AxMCQ0EWhhcNMTkwMzA1MTYwMzE4WhcNMjExMTI4MTYwMzIyWjANMQswCQYDVQQD
... EwJDQTCCAAIwDQYJKoZIhvcNAQEBBQADggGPADCCAYoCggGBAJmOTcw0mfZHwZQG
... K0QM8d0d38x5ABO45sxgiwx5SRwg0jC32Zpqc+b+JyNMH14IUHMYIoxi fLEDMtKv
... 0Lg77ARH37cyuqdIDsMkVXI//mgbHx6Qg8Wry0SkGJPby9jRwutz2G49ZtipmrRu
... zXvRjEHRBbfyqvjZmGc2A0Nc1Bp981ViTMWa3BKG9Ym20Tr/PtJpxvYnb85H89fs
... bfjVzQbyyIDFoTmnoaykBzMRGtxzjW/BUL3IzTvHTjFdHzJh7i80KKyLyepc573p
... b1uTWJJ8Sg8nS46tAU18G+7Y4pYMYh3gGEN9VuiPFV/vzWA7h5dELGOQe3tzSDSS
... 6XnILvQw1yN2R9LQ9Z08Xl8pEiJ/pwGfcBvIWPHPJDH8TnH3ZcvVwQnt98YwbmUx
... HGYR+2cFP+S6sTvw2ccvz4uENfKVstYTeVRrRHdTpHK7dzUWEU9UAWPpXOuflV69
... Zr6M3fBrBmDUrVaL864kPoDiCMNhtGDhU+Q3nQsvFH8HTm3zwIDAQABO0MwQTAP
... BgNVHRMBAf8EBTADAQH/MA8GA1UdDwEB/wQFAwMHBAAwHQYDVR0OBByEFF6FET3m
... 9NiPfbYqwF60m3yGTWXXMA0GCSqSIB3DQEBcWUAA4IBgQAHWozkh382EAI7i0wW
... CG94WJbxTTNnwA2e6FwQ0hSItr4RnfzeHm/DbmfNYlRKYAqGIsjzGLmWz+NozqOg
... Q6qK+RvGjr70zAXuygtrRzi32xuWbAijyx03VRv/FH91F8gf3plR3cNiAhVAW+ef
... xHiGzTrZh8E1HrrIJRj+uoQx66zxxIMZ8nEckxoqs0jFxyKY4/7sQ9mQAonlSQg9b
... Y+gJUecbt5Ff2SSyiUCM6XN0FuU/rXglrblscPdFUzeyX24TxWI36qtqYqmJff+d
... aniGfJlJ49Sg3iIoia3zXq7LTl4ZefSgRhb6V6eDzwX1vhx0005pGQRwAzW0wBaT
... k+IevZtdlKrEhycvrEoxSH70lPfgHBVJ5QrP80KkBR5WVa0fcQL/n+703ARepPH9
... Nj0Av+HazTPzVDIZwQg+fjftVZtR4CrlaeHGxZy8Tj3SkgiX5S0kTrb0nSHSwoTA
... taiY8eM9lD3siHbuGl/JT3wC8FXvq/cMhougM8NgzgeQ0Zc=
... -----END CERTIFICATE-----"
vrouters running tls#! certificate "-----BEGIN CERTIFICATE-----
... MIIDoTCCAgmgAwIBAgIIXH6fERmi2VkwDQYJKoZIhvcNAQELBQAwDTELMAkGA1UE
... AxMCQ0EWhhcNMTkwMzA1MTYwODUxWhcNMjExMTI4MTYwODUxWjARMQ8wDQYDVQQD
... EwZjbGllbnQwggeiMA0GCSqSIB3DQEBAQUAA4IBDwAwggEKAoIBAQC465HIizN9
... 3qzopW7tD57FjkccmmySPVHwM3dK1dZytS11ChCPf9rGAXcqagnXFGsjIoKULCWV
... 6eUMA6VQTW4XXIR1dn+x3pfgEp9of/6DR1dzK4UCpXrx4UIKQtDQ1R6UQ8QV1BaI
... ZNvR5Xl1HD/sS8LUcw8xGilKi+M0x6aPSJAtxoXgX2gn0w3Qn/SzxbCztNizPZO4
... Bk+YEVs6/vK2uyy87tISIkud2HhbxUkckySLawDZxbHr0xTgwc f4jz05GDXMokGx
... dRGVRNeRHfnDS9PZQoyvFeHKmsIrf8VqcMf6qtPGpjBnCm6WDq/rjosD1ZahQ1a7
... MXso0xg314MJAgMBAAGjgYAwfjAMBgNVHRMBAf8EAjAAMB0GA1UdJQQWMBQGCCsG
... AQUFBwMCMCBggrBgEFBQcDATAPBgNVHQ8BAf8EBQMDB6AAMB0GA1UdDgQWBRRrHgQ2
... 0vRIncZcD9MKUa6cFUSghjAfBgNVHSMEGDAWgBREhRE95vTYj322KsBetJt8hk8F
... 1zANBkgqhkig9w0BAQsFAA0CAYEABiDq/MS/YdXiNDE4lcE5A1qy3+S9WjPxs0q1

```

(continues on next page)

(continued from previous page)

```

... zQjr0cN/v/KvEg9S18dddgt1HTtdl/Wx57JjrmWymecM5E/HSU/3sxWNHSGjpsn
... gG4621jwrDWFvuzJHh1jJQyvNa6q++KmI1/Ubd9vL+g07Ity2zsRY4vxw6nmDfr5
... Vw6M18zh8wD6051Jm1AdR20518QfDNhm1mRdAuBacBO00J9/fC0zouOxgSy1W/ha
... 2PUJNf4nxNkQBngfMHKzf/fTmzticQ54Js/LoTkOCEBbhqajJ//eE6Bx5CIauJN
... dU06vfryih4wZ5rqsVk57i6lU1jBikvNnrai68MRzst4NUJBi0GVACfQv9efnyEM
... L03XAMMJUGZnvsrQbVZT1vJnfFehlrxgdXP8c9jiXSFnjZ7SjptxxQzOodaE+2jN
... 7KCJsizW4miGQyyeBoiQlIF9kjLT5kF41acACTzuPlDx0Lo00G3CB4APDvc8xWo
... J+z2SxQrkTVB5AHT7DyC+Zfjmnu6
... -----END CERTIFICATE-----"
vrouters running tls#! private-key "-----BEGIN RSA PRIVATE KEY-----
... MIIEpAIBAACAQEAuOuRyIszfd6s6KVu7Q+exY5HHJpskj1R1pt3StXWcrUpdQoQ
... j3/axgF3KmoJ1xRrIyKClCwllenlDGulUE80F1yEdXZ/sd6XxhKfaH/+g0dXcyuF
... AqV68eFCCkLQ0NUelEPEFdQWiGtB0eV5ZRw/7EvC1HMPMRopSovjNMemj0iQLcaF
... 4F9oJ9MN0J/0s8Wws7TysZ2TuAZPmBf0v7ytrssv07SEiJLndh4W8VJHJMki2sA
... 2cWx69MU4MHH+I89ORg1zKJBSXURLUTXkr35w0vT2UKMrxXhyprCK3/FanDH+qrT
... xqYwZwpulg6v646LA9WwOUNWuzF7KDsYN9eDCQIDAQABAoIBAHWjhw6pX4yHiEBI
... XhT5huvu41ZS9xbhY5q/NfirSM2YalNGn9ppX+bvL7wP0Uq+dpnXbnKM0yxXq5sH
... MBey8yfxd2KyI/G/xZYAAuCz7FfnfMZrvSY918TgpH6amvT/X4C6y5eHYP5MC3uw
... HFYybogIel1lBRkbp4EBFP2StWcYkQd2k7kEAhbk68IKzFLgDf9o6RL8/uSFHVds
... K6946+LRfu0KmMP6Qmf12pGdKwKPiTy1VI68SVwQBINLuN0tLPx5zgm1E9wGBghq
... FgB0wHt0vjF00eql+sjc7MLKv6iR56zDZupnv4rPnz5U9vCv83ApY4BcCmkFP1wd
... A7oPMCECgYEA60Tdn8A9dG58/jppEdhX0FQSOZjyK4tYMZVdLhg2TU9+0StbePe
... Qf0p8pc/7RtiOwyNu2fpJz3Q5vgfAlUPxSXkxukNfb9uCmAjNmKfYWMtRff/maFU
... E+Vei3MhG7NeR1SfcUEi0zUwW0hhpoIEdrRtWnD5fpWkzdkPh2bT8c0CgYEAy0Q/
... W2A9o3cW8qZHLA3nNT8hoT06v6hKcVvFPKwyG8q14VYCCYjaXHvxY0sXtweV+2yM
... 8v4sdn0GeVcJUFBri8NjYBTnuRtzTiZfbMDsR7QPwiIf7hyRaQF1KTBU0+givGOp
... XRUA97FUNHkyZUKwyWw8neG+tgOMx28ULz20Ci0CgYEA2bdqCp+T9DmVjr/5Gzwn
... hr6TYTMPwUEi5r9CkBT1ZNjjEoyHXJ2S3zmeB0zh0/Svhegcbz+atLaTHfiCdJm0
... XmcoUdL4a7+TTVuuAQ9V3txjWfjrukkQl1AXzjHkK/DyQcQ7r0noy6sAAAnQT+pn
... 5diSCeRnOLEIGe97FudH51kCgYBBhGn3hfnYKpaW98myixivLP41/pplFFWKWj4s
... TESKeLMnApX9hML9dGXf33pxYFyTgdWcrRi.fyITBr7As1v8TOYr5EUPvgk2ULwIr
... B7QhGITLyjwIf+T0t82PzSgZdyVbG7SHcDoVBG9jynzX7rsU8XJIYW8bZ3QFBGS5
... JWZWsQKBgQDDan2dn5URjua5zfJBBt4q92bMbkHLZIzQYPrsPv4v0XzVn6Y13Sy8
... uFmYrar4A0HQhpabIvH6MNpgJHK04g1ZALQrq3JIO+wpq6Mf4wyOXTANvLZCHjm/
... LhdDVcUs1nlM6zofsghgiYAcXQmZdDOrsv7PO0g54eOY0/8d2yRCJQ==
... -----END RSA PRIVATE KEY-----"
vrouters running tls#! server-authentication name server.example.org server-backup.
↪example.org
vrouters running tls#

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouters> show config xml absolute vrf main logging syslog
<config xmlns="urn:6wind:vrouter">

```

(continues on next page)

(continued from previous page)

```

<vrf>
  <name>main</name>
  <logging xmlns="urn:6wind:vrouter/logging">
    <syslog>
      <enabled>true</enabled>
      <remote-server>
        <host>10.125.0.2</host>
        <protocol>tcp</protocol>
        <port>514</port>
        <log-filter>
          <facility>any</facility>
          <level>
            <greater-or-equal>error</greater-or-equal>
          </level>
        </log-filter>
        <log-filter>
          <facility>kernel</facility>
          <level>
            <equal>any</equal>
          </level>
        </log-filter>
        <log-filter>
          <facility>auth</facility>
          <level>
            <equal>emergency</equal>
          </level>
        </log-filter>
        <log-filter>
          <facility>mail</facility>
          <level>
            <equal>none</equal>
          </level>
        </log-filter>
      </remote-server>
    <tls>
      <enabled>true</enabled>
      <server-authentication>
        <name>
          <name>server.example.org</name>
          <name>server-backup.example.org</name>
        </name>
      </server-authentication>
      <ca-certificate>-----BEGIN CERTIFICATE-----
MIID3zCCAkegAwIBAgIIXH6dxQIfVrcwDQYJKoZIhvcNAQELBQAwdTElMAkGA1UE

```

(continues on next page)

(continued from previous page)

```

AxMCQ0EwHhcNMTkwMZA1MTYwMzE4WhcNMjExMTI4MTYwMzIyWjANMQswCQYDVQQD
EwJDQTCCAaIwDQYJKoZIhvcNAQEBBQADggGPADCCAYoCggGBAJmOTcw0mfZHwZQG
K0QM8d0d38x5AB045sxgiwx5SRwg0jC32Zpqc+b+JyNMH14IUHMYIoxi fLEDMtKv
0Lg77ARH37cyuqdIDS MkVXI//mgbHx6Qg8Wry0SkgJPby9jRwutz2G49ZtipmrRu
zXvRjEHRBbfyqvjZmGc2A0Nc1Bp981ViTMWa3BKG9Ym20Tr/PtJpxvYnb85H89fs
bFjVzQbyyIDFoTmnoaykBzMRGtxzjW/BUL3IzTvHTjFdHzJh7i80KKyLyepc573p
b1uTWJJ8Sg8nS46tAU18G+7Y4pYMYh3gGEN9VuiPFV/vzWA7h5dELGOQe3tzSDSS
6XnILvQW1yN2R9LQ9ZO8X18pEiJ/pwGfcBvIWPHPJDH8TnH3ZcvVwQnt98YwbmUx
HGYR+2cfP+S6sTvW2ccvz4uENfKVstYTeVRRhDTPHK7dzUWEU9UAWPpXOUfLV69
Zr6M3fBrBmDUrVaL864kPoDiCMNhtGDhU+Q3nQsvFH8HBTm3zwIDAQABo0MwQTAP
BgNVHRMBAf8EBTADAQH/MA8GA1UdDwEB/wQFAwMHBAAwHQYDVRR0OBByEFF6FET3m
9NiPfbYqWf60m3yGTWXXMA0GCSqGSIb3DQEBCwUAA4IBgQAHWozkh382EAI7i0wW
CG94WJbxTTNnwA2e6FWqOhSitr4RnfzeHm/DbmfNYlRKYAqGIsjzGLmWz+NozqOg
Q6qK+RvGjr70zAXuygtRRzi32xuWbAijyx03VRv/FH91F8gf3plR3cNiAhVAW+ef
xHiGzTrZh8E1HrrIJRj+uoQx66zxkIMZ8nEckxoqs0jFxyK4/7sQ9mQAonlSQg9b
Y+gJUecBT5Ff2SSyiUCM6XN0FuU/rXglrblscPdFUZeyX24TxWI36qtqYqmJff+d
aniGfJlJ49Sg3iIoia3zXq7LTl4ZEfSgRhb6V6eDzwX1vhx0005pGQRwAzwOwBaT
k+IevZtdlKrEhycvrEoxSH70lPfgHBVJ5QrP80KkBR5WVa0fcQL/n+703ARepPH9
NjOAv+HazTPzVDIZwQg+fjftVZtR4CRLaeHGxZy8Tj3SkgiX5S0kTrb0nSHSwoTA
taiY8eM9lD3siHbuGl/JT3wC8FXvq/cMhougM8NgzgeQ0Zc=
-----END CERTIFICATE-----</ca-certificate>
<certificate>-----BEGIN CERTIFICATE-----
MIIDoTCCAgmgAwIBAgIIXH6fERmi2VkwDQYJKoZIhvcNAQELBQAwdTElMAkGA1UE
AxMCQ0EwHhcNMTkwMzA1MTYwODUxWhcNMjExMTI4MTYwODUxWjARMQ8wDQYDVQQD
EwZjbGllbnQwggeiMa0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC465HIizN9
3qzopW7tD57FjkccmmySPVHwM3dK1dZytSl1ChCPf9rGAXcqagnXFGsjIoKULCWV
6eUMA6VQTW4XXIR1dn+x3pfGep9of/6DR1dzK4UCpXrx4UIKQtDQ1R6UQ8QV1BaI
ZNVr5X1lHD/sS8LUcw8xGilkI+M0x6aPSJAtxoXgX2gn0w3Qn/SzxbCztNizPZO4
Bk+YEVs6/vK2uyy87tISIkud2HhbxUkckySLawDZxbHr0xTgwcF4jz05GDXMokGx
dRGVRNeRHfnDS9PZQoyvFeHKmsIrf8VqcMf6qtPGpjBnCM6WDq/rjosD1ZahQ1a7
MXso0xg314MJAgMBAAGjYAwfjAMBgNVHRMBAf8EAJAAMB0GA1UdJQVMBQGCCsG
AQUFBwMCAgggrBgEFBQcDATAPBgNVHQ8BAf8EBQMDB6AAMB0GA1UdDgQWBRRHgQ2
0vRIncZcD9MKUa6cFUSghjAfBgNVHSMEGDAWGBRehRE95vTYj322KsBetJt8hk8F
1zANBqkqhkiG9w0BAQsFAAOCAQEABiDq/MS/YdXiNDE4lce5A1qy3+S9WjPxs0ql
zQjr0cN/v/KvEg9S18dddtg1HTtdl/Wx57JjrmWymecM5E/HSU/3sxWNHSGjPJsN
gG4621jwrDWFvuzJHh1jJQyvNa6q++KmI1/UbD9vL+g07Ity2zsRY4vxw6nmDfr5
Vw6M18zh8wD6051Jm1AdR20518QfDNhm1mRdAuBacBO00J9/fc0zou0xgSy1W/ha
2PUJNf4nxNkQBngfMHKzf/fTmzticQ54Js/LoTkOCEBbhqapjJ//eE6Bx5CIauJN
dU06vfryih4wZ5rqsVk57i6lU1jBikvNnrai68MRzst4NUJBi0GVACfQv9efnyEM
L03XAMMJUGZnvsrQbVZT1vJnfFehlrxgdXP8c9jiXSFnjZ7SjptxxQzOodaE+2jN
7KCJsizW4miGQYyBoiQlIF9kjlT5kF41acACTzuPlDx0Lo00G3CB4APDvc8xWo
J+z2SxQrktVB5AHT7DyC+Zfjmn6
-----END CERTIFICATE-----</certificate>
<private-key>-----BEGIN RSA PRIVATE KEY-----

```

(continues on next page)

(continued from previous page)

```

MIIEpAIBAACKAQEAuOuRyIszfd6s6KVu7Q+exY5HHJpskj1R1pt3StXWcrUpdQoQ
j3/axgF3KmoJ1xRrIyKClCwlllenlDGulUE80F1yEdXZ/sd6XxhKfaH/+g0dXcyuF
AqV68eFCCKLQ0NUelEPEFdQWiGTb0eV5ZRw/7EvC1HMPMRopSovjNMemj0iQLcaF
4F9oJ9MN0J/0s8Wws7TYsz2TuAZPmBFb0v7ytrssv07SEiJLndh4W8VJHJMki2sA
2cWx69MU4MHH+I89ORg1zKJBSXURlUTXkr35w0vT2UKMrxXhyprCK3/FanDH+qrT
xqYwZwpulg6v646LA9WwoUNWuzF7KDsYN9eDCQIDAQABAoIBAHWjhw6pX4yHiEBI
XhT5huvu41ZS9xbhY5q/NFirSM2YaInGn9pqX+bvL7wP0Uq+dpnXbnKM0yxXq5sH
MBey8yfxd2KyI/G/xZYAauCz7FnfnMZrvSY918TgpH6amvT/X4C6y5eHYP5MC3uw
HFYybogIel1lBRkbp4EBFP2StWcYkQd2k7kEAhbk68IKzFLgDf9o6RL8/uSFHVds
K6946+LRfu0KmMP6QmfI2pGdKwKPiTyLVI68SVwQBINLuN0tLPx5zgm1E9wGBghq
FgB0wHt0vjF00eql+sjc7MLKv6iR56zDZupnv4rPnz5U9vCv83ApY4BcCmkFP1wd
A7oPMcECgYEA60TdIn8A9dG58/jppEdhXOFQSOZjyK4tYMZVdLhg2TU9+0StbePe
Qf0p8pc/7RtiOwyNu2fpJz3Q5vgfAlUPxSXkxukNfb9uCmAjNmKfYWMtRff/maFU
E+Vei3MhG7NeR1SfcUEi0zUwW0hhpoIEdrRtWnD5fpWkzdkPh2bT8c0CgYEAy0Q/
W2A9o3cW8qZHLA3nNT8hoT06v6hKcVvFPKwyG8q14VYCCYjaXHvxY0sXtweV+2yM
8v4sdn0GeVcJUFBri8NjYBTnuRtzTiZfbMDsR7QPwiIf7hyRaQF1KTBUO+givGop
XRUA97FUNHkyZUKwyWw8neG+ tqOMx28ULz20Ci0CgYEA2bdqCp+T9DmVjr/5Gzwn
hr6TYTMPwUEi5r9CkBT1ZNjjEoyHXJ2S3zmeB0zh0/Svhegcbz+atLaTHfiCdJm0
XmcoUdL4a7+TTVuuAQ9V3txjWFjrukkQ11AXzjHkK/DyQcQ7r0noy6sAAAnQT+pn
5diScErnOLEIGe97FudH51kCgYBBhGn3hfnYKpaW98myixivLP41/pplFFWKWj4s
TESKeLMnApX9hML9dGXF33pxYFyTgdWcrRifyITBr7As1v8TOYr5EUPvgk2ULwIr
B7QhGITLyjwIf+T0t82PzSgZdyVbG7SHcDoVBG9jynzX7rsU8XJIYw8bZ3QFBGS5
JWZWsQKBgQDDan2dN5URjua5zfJBBt4q92bMbkHLZIzQYpRsPv4vOXzVn6Y13Sy8
uFmYrar4AOHQhpabIvH6MNpgJHK04g1ZALQrq3JIO+wpq6Mf4wyOXTANvLZCHjm/
LhhDVcUs1nlM6zofsgghiYAcXQmZdD0rsv7P00g54e0Y0/8d2yRCJQ==
-----END RSA PRIVATE KEY-----</private-key>
</tls>
</syslog>
</logging>
</vrf>
</config>

```

### 3.1.5 Network interfaces

#### Interface types

#### Overview

Turbo Router supports physical and logical network interfaces.

Interfaces are configured within the VRF they belong to. Interface names are given by the user when creating the interfaces.

The general syntax for creating an interface is as follows:

```
running vrf main# interface TYPE NAME
```

where NAME is the name of the interface and TYPE can be:

**physical** to create an *Ethernet interface*

**gre** to create a *GRE interface*

**ipip** to create an *IPv4 and IPv6 tunneling*

**vlan** to create a *VLAN interface*

**vxlan** to create a *VXLAN interface*

**lag** to create a *LAG interface*

**bridge** to create a *bridge interface*

**loopback** to create a *loopback interface*

**system loopback** to create a *system loopback (lo) interface*

**veth** to create a *veth interface*

## Ethernet

### Overview

Ethernet interfaces represent NICs in the management system.

To create an Ethernet interface, use the `interface physical` command in a VRF.

```
vrouter running config# vrf main
vrouter running vrf main# interface physical eth0
vrouter running physical eth0#!
```

The exclamation mark at the end of the prompt means that the configuration is incomplete. This is because Ethernet interfaces require a port identifier.

The matching between port identifiers and PCI identifiers of Network Interface Cards is displayed system using the `show state / network-port` command.

```
vrouter running physical eth0#! show state / network-port
network-port pci-b0s3
  pci-bus-addr 0000:00:03.0
  vendor "Red Hat, Inc"
  model "Virtio network device"
  mac-address 52:54:00:12:34:57
  interface eth0
  ..
```

(continues on next page)

(continued from previous page)

```

network-port pci-b0s9
  pci-bus-addr 0000:00:09.0
  vendor "Red Hat, Inc"
  model "Virtio network device"
  mac-address de:ad:de:01:02:03
  interface eth1
  ..
network-port pci-b0s8
  pci-bus-addr 0000:00:08.0
  vendor "Red Hat, Inc"
  model "Virtio network device"
  mac-address 52:54:00:12:34:56
  interface eth2
  ..
vrouters running physical eth0#!

```

Use the `port` command to associate a port identifier to the Ethernet interface:

```
vrouters running physical eth0#! port pci-b0s8
```

After committing the configuration, we can fetch the state of the interface using the following command:

```

vrouters running physical eth0# commit
vrouters running physical eth0# show state / vrf main interface physical eth0
physical eth0
  mtu 1500
  enabled true
  port pci-b0s8
  oper-status UP
  counters
    in-octets 0
    in-unicast-pkts 0
    in-discards 0
    in-errors 0
    out-octets 0
    out-unicast-pkts 6
    out-discards 0
    out-errors 0
  ..
  ipv6
    address fe80::a00:27ff:fea9:a96e/64
  ..
  ethernet
    mac-address 08:00:27:a9:a9:6e

```

(continues on next page)

(continued from previous page)

```

..
..
vrouters running physical eth0#

```

The same configuration can be applied using the following NETCONF XML configuration:

```

vrouters> show config xml absolute vrf main interface physical eth0
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <physical>
        <name>eth0</name>
        <enabled>true</enabled>
        <ipv4>
          <enabled>true</enabled>
        </ipv4>
        <ipv6>
          <enabled>true</enabled>
        </ipv6>
        <port>pci-b0s8</port>
      </physical>
    </interface>
  </vrf>
</config>

```

## Control Plane Protection

*Control Plane Protection* is a software mechanism that reduces the risk of dropping control packets. It can be enabled on physical interfaces when the fast path is running. See the fast path *control plane protection* section for details.

### See also:

The *command reference* for details.

## GRE

### Basic configuration

GRE protocol provides a simple and general mechanism to encapsulate a network layer protocol in another network layer protocol. It is defined in RFC 2784.

This interface is point to point or point to multipoint. So its configuration is different from ethernet interfaces (arp/ndp, dhcp are not available).

To configure GRE, enter the context `interface type gre` from the VRF in which you plan to define a GRE interface.

Here is an example of a point to point GRE named `tunnel1`, with connecting the local address `1.1.1.1` and the remote address `2.2.2.2`:

```
vrouter running vrf main# interface gre tunnel1
vrouter running gre tunnel1#! local 1.1.1.1 remote 2.2.2.2
vrouter running gre tunnel1# commit
```

Let's fetch the state after committing this configuration:

```
vrouter running vrf main# interface gre tunnel1
vrouter running gre tunnel1# show state
gre tunnel1
  remote 2.2.2.2
  enabled true
  oper-status UP
  mtu 1476
  local 1.1.1.1
  counters
    in-octets 0
    out-octets 0
    in-errors 0
    in-unicast-pkts 0
    in-discards 0
    out-unicast-pkts 0
    out-errors 0
    out-discards 0
    ..
  ipv6
    address fe80::200:5efe:101:101/64
    ..
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```

vrouter running gre tunnel1# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <gre xmlns="urn:6wind:vrouter/gre">
        <name>tunnel1</name>
        <enabled>true</enabled>
        (...)
        <local>1.1.1.1</local>
        <remote>2.2.2.2</remote>
      </gre>
    </interface>
  </vrf>
</config>

```

## Link VRF

A GRE interface may perform cross-VRF, i.e change the VRF of encapsulated and decapsulated packets:

```

vrouter running vrf main# interface gre tunnel1
vrouter running gre tunnel1# link-vrf wan

```

The link VRF is the VRF of encapsulated packets. The interface VRF is the VRF of output packets before encapsulation and of input packets after decapsulation.

## GRE key

The GRE key is an extension defined in RFC 2890. It is an optional 32 bit field that enables to identify an individual traffic flow or service within a GRE tunnel.

When using this feature, each individual flow/service is processed by a different GRE interface, identified with the key assigned to the flow/service.

An optional output key may be assigned to a GRE interface. If set, GRE packets output by this interface will have a key field with the configured value:

```

vrouter running vrf main# interface gre tunnel1
vrouter running gre tunnel1# key output 5

```

An optional input key may be assigned to a GRE interface. If set, only GRE packets with a key field set to this value will be processed by this interface. If unset, only GRE packets without a key field will be processed by this interface.

```
vrouter running gre tunnel1# key input 2
```

key both assigns the same value for the input and output keys. It is overridden if key input or key output is specified:

```
vrouter running gre tunnel1# key both 3
```

The use of input and output keys is independent: it is possible to assign an output key without assigning an input key, and vice versa.

The tuple (local, remote, link-vrf, key input) must be unique among all GRE interfaces, whatever their vrf.

## GRE multipoint

A point to multipoint GRE is simply defined by letting the remote address empty. As the remote address is not specify it is needed to specify neighbors for routing operations.

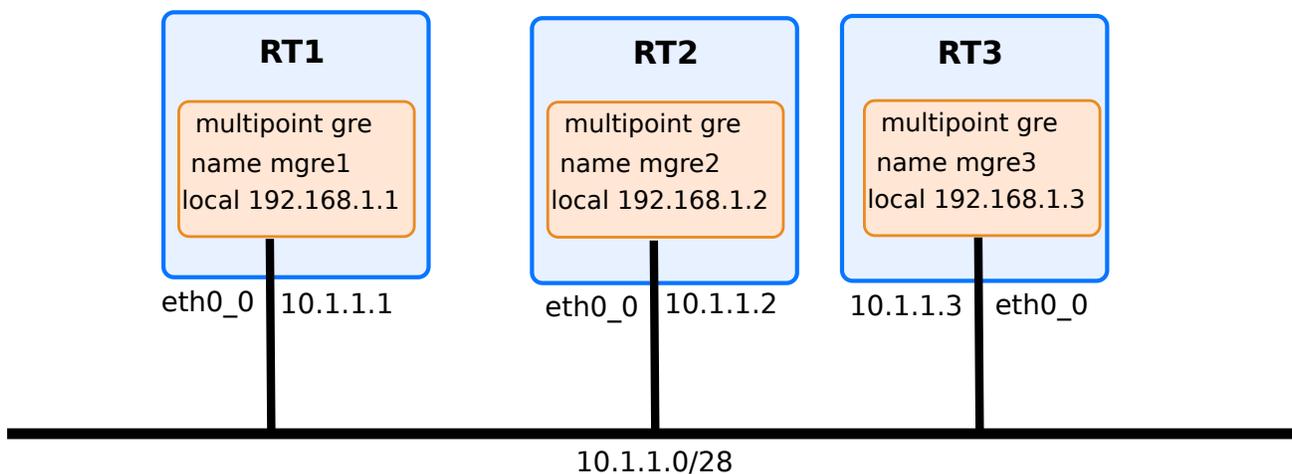


Fig. 1: GRE multipoint configuration illustration with 3 GRE tunnels

The configuration of point to multipoint GRE named gre1 is:

```
vrouter running vrf main# interface gre mgre1
vrouter running gre mgre1#! local 10.1.1.1
vrouter running gre mgre1# enabled true
vrouter running gre mgre1# link-interface eth0_0
vrouter running gre mgre1# ipv4 address 192.168.1.1/24
vrouter running gre mgre1# ipv4 neighbor 192.168.1.2 link-layer-address 10.1.1.2
vrouter running gre mgre1# ipv4 neighbor 192.168.1.3 link-layer-address 10.1.1.3
vrouter running gre mgre1# commit
```

The same configuration can be applied to point to multipoint GRE named gre2 and gre3 to connect the three routers with a multipoint GRE.

Let's fetch the state after committing this configuration:

```
vrouter running vrf main# interface gre mgrel
vrouter running gre mgrel# show state
gre mgrel
  mtu 1476
  promiscuous false
  enabled true
  ipv4
    address 192.168.1.1/24
    neighbor 192.168.1.2 link-layer-address 10.1.1.2 state permanent
    neighbor 192.168.1.3 link-layer-address 10.1.1.3 state permanent
    ..
  ipv6
    address fe80::5efe:a01:101/64
    ..
  link-interface eth0_0
  local 10.1.1.1
  oper-status UP
  counters
    in-octets 0
    in-unicast-pkts 0
    in-discards 0
    in-errors 0
    out-octets 0
    out-unicast-pkts 0
    out-discards 0
    out-errors 0
    ..
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running gre mgrel# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <gre xmlns="urn:6wind:vrouter/gre">
        <name>mgrel</name>
        <enabled>true</enabled>
        <ipv4>
          <enabled>true</enabled>
```

(continues on next page)

(continued from previous page)

```

    <address>
      <ip>192.168.1.1/24</ip>
    </address>
    <neighbor>
      <ip>192.168.1.2</ip>
      <link-layer-address>10.1.1.2</link-layer-address>
    </neighbor>
    <neighbor>
      <ip>192.168.1.3</ip>
      <link-layer-address>10.1.1.3</link-layer-address>
    </neighbor>
  </ipv4>
  <ipv6>
    <enabled>>true</enabled>
  </ipv6>
  <key/>
  <local>10.1.1.1</local>
  <link-interface>mgmt0</link-interface>
</gre>
</interface>
</vrf>
</config>

```

**See also:**

The *command reference* for details.

**IPv4 and IPv6 tunneling**

Tunneling is a widespread technique used in networking, to resolve many problems: IPv4 / IPv6 migration, Virtual Private Networks, routing. It consists in encapsulating a packet into a new layer 3 packet, by appending an IP header. 6WIND Turbo Router provides several techniques to tunnel IP packets into new IP packets (the inner and outer IP versions may differ).

Tunneling techniques create a virtual layer 2 link (called a tunnel) between the source and destination of the encapsulating packets, and hide the network topology between these two endpoints, as if the two endpoints were directly connected. Therefore, 6WIND Turbo Router creates a logical point-to-point interface, that appears in the list of interfaces and that can be used by other functions, notably routing.

There are 4 different types of tunnel:

- 4in4. IPv4 in IPv4 Configured Tunnels encapsulates IPv4 traffic in an explicit IPv4 tunnel.
- 6in4. An IPv6 in IPv4 configured tunnel encapsulates IPv6 traffic in an explicit IPv4 tunnel.
- 4in6. IPv4 in IPv6 Configured Tunnels encapsulates IPv4 traffic in an explicit IPv6 tunnel. That could be

useful to simulate VLANs. That could be useful for the interconnection of IPv4 clouds on an IPv6 native service

- 6in6. IPv6 in IPv6 Configured Tunnels encapsulates IPv6 traffic in an explicit IPv6 tunnel.

Here is an example of a 4in6 tunnel named `tun4in6` in VRF `main`, linked to underlying interface named `eth0`.

```
vrouter running vrf main# interface ipip tun4in6
vrouter running ipip tun4in6#! local fd00:125::1 remote fd00:125::2 link-interface eth0
vrouter running ipip tun4in6# ipv4 address 192.168.0.1 peer 192.168.0.2
vrouter running ipip tun4in6# commit
```

The tunnel interface is configured as soon as the provided `eth0` is configured in VRF `main`.

Let's fetch the state after committing this configuration:

```
vrouter running vrf main# interface ipip tun4in6
running ipip tun4in6# show state
ipip tun4in6
  mtu 1452
  enabled true
  ipv4
    address 192.168.0.1 peer 192.168.0.2
  ..
  ipv6
    address fe80::7cb3:5fff:feb7:e3af/64
  ..
  local fd00:125::1
  remote fd00:125::2
  link-interface eth0
  oper-status UNKNOWN
  counters
    in-octets 0
    in-unicast-pkts 0
    in-discards 0
    in-errors 0
    out-octets 0
    out-unicast-pkts 0
    out-discards 0
    out-errors 0
  ..
..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running config# show config xml absolute vrf main interface ipip tun4in6
<config xmlns="urn:6wind:vrouter">
```

(continues on next page)

(continued from previous page)

```

<ha xmlns="urn:6wind:vrouter/ha"/>
<vrf>
  <name>main</name>
  <interface xmlns="urn:6wind:vrouter/interface">
    <ipip xmlns="urn:6wind:vrouter/ipip">
      <name>tun4in6</name>
      <enabled>true</enabled>
      <ethernet/>
      <ipv4>
        <enabled>true</enabled>
        <address>
          <ip>192.168.0.1</ip>
          <peer>192.168.0.2</peer>
        </address>
      </ipv4>
      <ipv6>
        <enabled>true</enabled>
      </ipv6>
      <local>fd00:125::1</local>
      <remote>fd00:125::2</remote>
      <link-interface>eth0</link-interface>
    </ipip>
  </interface>
</vrf>
</config>

```

**See also:**

The *command reference* for details.

**VLAN**

Virtual Local Area Networks (VLAN) allows to divide a network into several logical networks domains. The standard 802.1Q protocol is used to add a tag identifier between 1 and 4094. VLAN stacking or QinQ is supported by simply binding the VLAN interface to another.

To configure VLAN, enter the context `interface type vlan` from the VRF in which you plan to define VLAN logical interface. The VLAN configuration is valid as soon as the VLAN ID is set and the bound interface is set.

Here is an example of VLAN named `vlan-blue` in VRF `main`, with a tag identifier `300` and bound to underlying interface named `eth0`:

```

vrouter running vrf main# interface vlan vlan-blue
vrouter running vlan vlan-blue#! vlan-id 300

```

(continues on next page)

(continued from previous page)

```
vrouters running vrf main# link-interface eth0
vrouters running vrf main# commit
```

The VLAN interface is configured provided eth0 is configured in VRF main.

Let's fetch the state after committing this configuration:

```
vrouters running vrf main# interface vrf main interface vrf main
vrouters running vrf main# show state
vrf main
  protocol 802.1q
  ethernet
    mac-address de:ad:de:01:02:03
    ..
  mtu 1500
  counters
    out-octets 0
    in-octets 0
    in-unicast-pkts 0
    out-unicast-pkts 9
    in-discards 0
    in-errors 0
    out-discards 0
    out-errors 0
    ..
  link-interface eth0
  oper-status UP
  enabled true
  ipv6
    address fe80::dcad:deff:fe01:203/64
    ..
  ..
  vlan-id 300
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouters> show config xml absolute vrf main interface vrf main interface vrf main
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <vrf xmlns="urn:6wind:vrouter/vrf">
        <name>vrf-main</name>
        <protocol>802.1q</protocol>
      </vrf>
    </interface>
  </vrf>
</config>
```

(continues on next page)

(continued from previous page)

```

    <vlan-id>300</vlan-id>
    <link-interface>eth0</link-interface>
    (...)
  </vlan>
</interface>
</vrf>
</config>

```

**See also:**

The *command reference* for details.

**Cross VRF setup**

By default, the link interface must be in the same VRF than the VLAN interface. However, a VLAN interface can bind a link interface which is located in another VRF: this type of setup is called *cross-vrf*.

To change the link VRF, set the *link-vrf*:

```

vrouters running vrf main# interface vlan vlan-green
vrouters running vlan vlan-green#! vlan-id 400
vrouters running vlan vlan-green#! link-interface eth0
vrouters running vlan vlan-green# link-vrf vrf1
vrouters running vlan vlan-green# commit

```

**VXLAN**

Virtual eXtensible Local Area Networks (VXLAN) is used to address the need for overlay networks within virtualized data centers accommodating multiple tenants.

To configure VXLAN, enter the context `interface` type `vxlan` from the VRF in which you plan to define VXLAN logical interface. The VXLAN configuration is valid as soon as the VXLAN ID is set.

Here is an example of VXLAN named `vxlan100` in VRF `main`, with a tag identifier `100` and linked to underlying interface named `eth0` using the multicast group `'239.0.0.8'`:

```

vrouters running vrf main# interface vxlan vxlan100
vrouters running vxlan vxlan100#! vni 100
vrouters running vxlan vxlan100# link-interface eth0
vrouters running vxlan vxlan100# group '239.0.0.8'
vrouters running vxlan vxlan100# commit

```

The VXLAN interface is configured provided `eth0` is configured in VRF `main`.

Let's fetch the state after committing this configuration:

```

vrouters running vrf main# interface vxlan vxlan100
vrouters running vxlan vxlan100# show state
vxlan vxlan100
  mtu 1450
  enabled true
  ethernet
    mac-address 36:22:c6:04:24:49
    ..
  ipv6
    address fe80::3422:c6ff:fe04:2449/64
    ..
  vni 100
  group 239.0.0.8
  link-interface eth0
  learning true
  gbp false
  dst 4789
  src-range
    49152
    65535
    ..
  oper-status UNKNOWN
  counters
    in-octets 0
    in-unicast-pkts 0
    in-discards 0
    in-errors 0
    out-octets 0
    out-unicast-pkts 8
    out-discards 0
    out-errors 0
    ..
  ..

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouters> show config xml absolute vrf main interface vxlan vxlan100
<config xmlns="urn:6wind:vrouter">
  <ha xmlns="urn:6wind:vrouter/ha"/>
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <vxlan xmlns="urn:6wind:vrouter/vxlan">
        <name>vxlan100</name>
        <enabled>true</enabled>

```

(continues on next page)

(continued from previous page)

```

<ethernet>
  <auto-negotiate>true</auto-negotiate>
  <enable-flow-control>>false</enable-flow-control>
</ethernet>
<ipv4>
  <enabled>true</enabled>
</ipv4>
<ipv6>
  <enabled>true</enabled>
</ipv6>
<learning>true</learning>
<gbp>>false</gbp>
<dst>4789</dst>
<src-range>
  <min>49152</min>
  <max>65535</max>
</src-range>
<vni>100</vni>
<link-interface>eth0</link-interface>
<group>239.0.0.8</group>
</vxlan>
</interface>
</vrf>
</config>

```

An alternative configuration is depicted below, which is using the benefits of EVPN technology. With BGP, as per chapter *BGP EVPN*, VXLAN interfaces are used to connect VPNs (Virtual Private Networks) between different sites. VXLAN configuration differs a bit compared with previous configuration. For instance, there is no need to configure any group, as it is up to BGP to handle which overlay traffic to convey to vxlan interface, and which destination IP (Internet Protocol) to apply to the underlay destination IP. Below configuration is an illustration on how VXLAN interface can be configured within EVPN configuration:

**rt1**

```

vrf custom1
  interface
    vxlan vxl11
      mtu 1550
      vni 11
      local 10.125.0.1
      link-vrf main
      learning true
      link-interface eth0

```

(continues on next page)

(continued from previous page)

```

    link-vrf main
    ..
  ..
interface
  bridge br11
  link-interface vx111
  mtu 1500
  ..
  ..

```

More information can be found on EVPN chapter in routing chapter, in BGP.

**See also:**

The *command reference* for details.

## LAG

Link Aggregation (LAG) allows to aggregate multiple network interfaces into a single logical “bonded” interface. It can provide an active backup service assisted with LACP (802.3ad) or a load balancing to increase the bandwidth.

Multiple modes are available for load balancing: round-robin, XOR on MAC address.

Multiple policies are available to select which part of the packet header will be used to compute the hash: L2, L3, L4, mix of L2 and L3, mix of L3 and L4, using either the outer or the most inner packet in case of encapsulation.

By default the MII link monitoring is activated and set to 100 ms. Disabling the MII link monitoring (set its value to 0) is not recommended, the link detection and failure can have poor performance.

To configure a LAG, enter the context `interface type lag` from the VRF in which you plan to define the LAG interface.

Here is an example of lag named `lag0` in VRF `main`, using `lacp` mode with a hash on L2+L3 header and a slow rate using two interfaces `eth0` and `eth1`.

```

running vrf main# interface lag lag0
running vrf main# mode lacp xmit-hash-policy layer2+3 lacp-rate slow
running lag lag0# link-interface eth0
running lag lag0# link-interface eth1
running lag lag0# commit

```

The lag interface is configured provided `eth0` and `eth1` are present in VRF `main`.

Let's fetch the state after committing this configuration:

```

running vrf main# interface lag lag0
running lag lag0# show state

```

(continues on next page)

(continued from previous page)

```

lag lag0
  mii-link-monitoring 100
  mode lacp
  xmit-hash-policy layer2+3
  mtu 1500
  lacp-rate slow
  oper-status DOWN
  link-interface eth0
  link-interface eth1
  enabled true
  ethernet
    mac-address f2:a2:6c:f2:9e:e4
    ..
  counters
    in-octets 0
    in-discards 0
    in-errors 0
    out-octets 0
    in-unicast-pkts 0
    out-errors 0
    out-discards 0
    out-unicast-pkts 0
    ..
  ..

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouter> show xml absolute config vrf main interface lag lag0
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <lag xmlns="urn:6wind:vrouter/lag">
        <name>lag0</name>
        <enabled>true</enabled>
        <link-interface>
          <slave>eth0</slave>
        </link-interface>
        <link-interface>
          <slave>eth1</slave>
        </link-interface>
        <ipv4>
          <enabled>true</enabled>
        </ipv4>
      </lag>
    </interface>
  </vrf>
</config>

```

(continues on next page)

(continued from previous page)

```

    <ipv6>
      <enabled>true</enabled>
    </ipv6>
    <mii-link-monitoring>100</mii-link-monitoring>
    <mode>lacp</mode>
    <xmit-hash-policy>layer2+3</xmit-hash-policy>
    <lacp-rate>slow</lacp-rate>
  </lag>
</interface>
</vrf>
</config>

```

**See also:**

The *command reference* for details.

**Bridge**

Bridge allows the connection of two separate networks as if they were a single network. It builds a database by inspecting the destination MAC address of packets flowing through the bridged interfaces: known destination is forwarded, unknown is broadcast to all other networks.

To configure a bridge, enter the context `interface type bridge` from the VRF in which you plan to define the bridge logical interface. The bridge configuration is valid as soon as the slave interfaces are set.

Here is an example of bridge named `br0` in VRF `main`, using two interfaces `eth0` and `eth1`.

```

vrouters running vrf main# interface bridge br0
vrouters running bridge br0# link-interface eth0
vrouters running bridge br0# link-interface eth1
vrouters running bridge br0# commit

```

The bridge interface is configured provided `eth0` and `eth1` are present in VRF `main`.

Let's fetch the state after committing this configuration:

```

vrouters running vrf main# interface bridge br0
vrouters running bridge br0# show state
bridge br0
  oper-status UNKNOWN
  enabled true
  mtu 1500
  link-interface eth0
  link-interface eth1
  ethernet

```

(continues on next page)

(continued from previous page)

```

        mac-address 9a:cb:9c:2e:fd:07
        ..
    counters
        in-octets 0
        out-octets 0
        in-errors 0
        in-unicast-pkts 0
        in-discards 0
        out-unicast-pkts 7
        out-errors 0
        out-discards 0
        ..
    ipv6
        address fe80::98cb:9cff:fe2e:fd07/64
        ..
    ..

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouters running config# show config xml absolute vrf main interface bridge br0
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <bridge xmlns="urn:6wind:vrouter/bridge">
        <name>br0</name>
        <enabled>>true</enabled>
        (...)
        <link-interface>
          <slave>eth0</slave>
        </link-interface>
        <link-interface>
          <slave>eth1</slave>
        </link-interface>
      </bridge>
    </interface>
  </vrf>
</config>

```

**See also:**

The *command reference* for details.

## Loopback

The main purpose of loopback interfaces is to provide one or more permanent addresses to a network device, regardless of which network interfaces are up. A loopback address is typically announced into the routing tables, and can therefore be used as a management address instead of a physical interface address. This is preferable since a loopback interface is independent from any physical interface and is, therefore, always available. This also enables to configure unnumbered point-to-point interfaces (for example with a PPPv4 server) A loopback address will typically be used in IPv4 packets. Finally, a prefix configured on a loopback interface can be used to announce some directly connected networks via dynamic routing protocols.

To configure loopback, enter the context `interface type loopback` from the VRF in which you plan to define a loopback logical interface.

```
vrouter running vrf main# interface loopback loop0
vrouter running loopback loop0# commit
```

Let's fetch the state after committing this configuration:

```
vrouter running vrf main# interface loopback loop0
vrouter running loopback loop0# show state
loopback loop0
  oper-status UP
  enabled true
  mtu 1500
  counters
    in-octets 0
    out-octets 0
    in-errors 0
    in-unicast-pkts 0
    in-discards 0
    out-unicast-pkts 0
    out-errors 0
    out-discards 0
  ..
  ethernet
    mac-address 26:16:54:8d:10:0a
  ..
  ipv6
    address fe80::2416:54ff:fe8d:100a/64
  ..
..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running loopback loop0# show config xml absolute
<config xmlns="urn:6wind:vrouter">
```

(continues on next page)

(continued from previous page)

```
<vrf>
  <name>main</name>
  <interface xmlns="urn:6wind:vrouter/interface">
    <loopback xmlns="urn:6wind:vrouter/loopback">
      <name>loop0</name>
      (...)
    </loopback>
  </interface>
</vrf>
</config>
```

**See also:**

The *command reference* for details.

---

**Note:** SVTI requires a Turbo IPsec Application License.

---

## SVTI

Secure Virtual Tunnel Interfaces are generic virtual interfaces ensuring IPsec transformation. They are used to configure route-based VPNs.

Each SVTI interface has its own SAD (Security Association Database) and SPD (Security Policy Database).

Unlike other tunnel interfaces, an SVTI interface is not a point-to-point interface between two specific gateways, the encapsulation addresses are determined by the SPs (Security Policies) and SAs (Security Associations) matched by the traffic.

These interfaces have an SVTI ID parameter to associate them to IPsec SAs/SPs. This ID must be unique per-VRF<sup>1</sup>.

To configure SVTI, enter the `interface` context in the desired VRF and type `svti` followed by the SVTI interface name. The interface name must start with `svti`. The configuration is valid as soon as the SVTI identifier is set.

Here is an example of an SVTI named `svti100` with an SVTI identifier `100`:

```
vrouters running vrf main# interface svti svti100
vrouters running svti svti100#! svti-id 100
vrouters running svti svti100# commit
```

The SVTI interface is configured and ready to be associated to an IKE VPN.

Let's fetch the state after committing this configuration:

---

<sup>1</sup> To be precise, the (link VRF, SVTI ID) pair must be unique, see paragraph *Cross-VRF*.

```

vrouters running vrf main# interface svti svti100
vrouters running svti svti100# show state
svti svti100
  mtu 1500
  promiscuous false
  enabled true
  ipv6
    address fe80::afb4:e94a:240a:23f3/64
    ..
  svti-id 100
  oper-status UNKNOWN
  counters
    in-octets 0
    in-unicast-pkts 0
    in-discards 0
    in-errors 0
    out-octets 0
    out-unicast-pkts 0
    out-discards 0
    out-errors 0
    ..
  link-interface lo
  ..

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouters> show config xml absolute vrf main interface svti svti100
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <svti xmlns="urn:6wind:vrouter/svti">
        <name>svti100</name>
        <enabled>true</enabled>
        <ipv4>
          <enabled>true</enabled>
        </ipv4>
        <ipv6>
          <enabled>true</enabled>
        </ipv6>
        <svti-id>100</svti-id>
      </svti>
    </interface>
  </vrf>
</config>

```

## Cross-VRF

Like other tunnel interfaces, an SVTI interface is defined in a VRF and is assigned a possibly different link VRF (the VRF of encapsulated packets). SVTI interfaces can therefore be used to do cross-VRF.

The (link VRF, SVTI ID) pair uniquely identifies an SVTI interface, regardless of the interface VRF.

The SPs, SAs and IKE configuration are located in the SVTI interface link VRF.

Here is an example of an SVTI located in vrf2 but with a link-vrf in vrf1:

```
vrouter running vrf vrf2# interface svti svti100
vrouter running svti svti100#! svti-id 100
vrouter running svti svti100# link-vrf vrf1
vrouter running svti svti100# commit
```

In this configuration, the clear traffic will be in vrf2 and the encrypted traffic in vrf1.

## SVTI templates

SVTI templates are models of SVTI interfaces used for dynamic SVTI interface creation. An IKE VPN may reference an SVTI template, so that an SVTI interface is created for each established IKE SA, and the negotiated SAs and SPs attached to it.

To configure an SVTI template, enter the `interface` context in the desired VRF and type `svti-template` followed by the SVTI template name.

Here is an example of an SVTI template named `svtitemp100`:

```
vrouter running vrf main# interface svti-template svtitemp100
vrouter running svti-template svtitemp100# mtu 1300
vrouter running svti-template svtitemp100# commit
```

The dynamically created SVTI interface names will start with `dsvti` and do not depend on the template name. Their SVTI ID is dynamically chosen by IKE.

### See also:

The *command reference* for details.

## System Loopback

The system loopback interface is the `lo` interface created by the system in every VRF. It cannot be configured, but it is advertised in the state:

```
vrouter> show state vrf main interface system-loopback
system-loopback lo
  mtu 65536
  enabled true
  ipv4
    address 127.0.0.1/8
    ..
  ipv6
    address ::1/128
    ..
  oper-status UP
  counters
    in-octets 37993
    in-unicast-pkts 192
    in-discards 0
    in-errors 0
    out-octets 37993
    out-unicast-pkts 192
    out-discards 0
    out-errors 0
    ..
  ..
```

### See also:

The *command reference* for details.

## veth

A usual way to connect VRF together is to use a `veth` interface. The `veth` interfaces are virtual Ethernet devices that are always created in interconnected pairs. They can act as tunnels between network namespaces.

`veth` interfaces are similar to `xvrf` interfaces, with the following differences:

- the MAC (Medium Access Control) address can be configured on `veth` interfaces
- `veth` interfaces are not flagged `NOARP`, meaning that ARP or NDP (Neighbor Discovery Protocol) resolution is done when sending an IP packet through it
- `veth` interfaces support IP configuration

**See also:**

*xvrf interfaces.*

Here is an example of configuration where veth interfaces connect two VRF.

```
vrouter running config# / vrf vr1
vrouter running vrf vr1# interface veth veth-to-vr2
vrouter running veth veth-to-vr2#! link-interface veth-to-vr1 link-vrf vr2
vrouter running veth veth-to-vr2#! ipv4 address 10.1.1.1/24
vrouter running veth veth-to-vr2#! / vrf vr2
vrouter running vrf vr2#! interface veth veth-to-vr1
vrouter running veth veth-to-vr1#! link-interface veth-to-vr2 link-vrf vr1
vrouter running veth veth-to-vr1#! ipv4 address 10.1.1.2/24
vrouter running veth veth-to-vr1#! commit
```

A YANG condition ensures that the binding of veth interfaces is consistent: the veth interfaces of a given pair must bind each other.

A route can then be added in vr2 to reach a network 10.100.0.0/16 through vr1:

```
vrouter running config# vrf vr2
vrouter running vrf vr2# routing static
vrouter running static# ipv4-route 10.100.0.0/16 next-hop 10.1.1.1
vrouter running static# commit
```

Let's fetch the veth state inside vr1 after committing this configuration:

```
vrouter running config# show state vrf vr1 interface veth
veth veth-to-vr2
  mtu 1500
  promiscuous false
  enabled true
  ipv4
    address 10.1.1.1/24
    ..
  ipv6
    address fe80::687e:84ff:fed1:cc6/64
    ..
  oper-status UP
  counters
    in-octets 738
    in-unicast-pkts 7
    in-discards 0
    in-errors 0
    out-octets 738
    out-unicast-pkts 7
```

(continues on next page)

(continued from previous page)

```

    out-discards 0
    out-errors 0
    ..
  ethernet
    mac-address 6a:7e:84:d1:0c:c6
    ..
  link-interface veth-to-vr1
  link-vrf vr2
  ..

```

The same configuration can be made using this NETCONF XML configuration.

```

vrouters running config# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>vr1</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <veth xmlns="urn:6wind:vrouter/veth">
        <name>veth-to-vr2</name>
        <enabled>true</enabled>
        <ipv4>
          <enabled>true</enabled>
          <address>
            <ip>10.1.1.1/24</ip>
          </address>
        </ipv4>
        <link-interface>veth-to-vr1</link-interface>
        <link-vrf>vr2</link-vrf>
        (...)
      </veth>
    </interface>
  </vrf>
  <vrf>
    <name>vr2</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <veth xmlns="urn:6wind:vrouter/veth">
        <name>veth-to-vr1</name>
        <enabled>true</enabled>
        <ipv4>
          <enabled>true</enabled>
          <address>
            <ip>10.1.1.2/24</ip>
          </address>
        </ipv4>

```

(continues on next page)

(continued from previous page)

```

    <link-interface>veth-to-vr2</link-interface>
    <link-vrf>vr1</link-vrf>
    (...)
  </veth>
</interface>
</vrf>
</config>

```

**See also:**

The *command reference* for details.

**XVRF**

A usual way to connect VRF together is to use a `xvrf` interface. The `xvrf` interfaces are virtual Ethernet devices that are always created in interconnected pairs. They can act as tunnels between network namespaces.

`xvrf` interfaces are similar to `veth` interfaces, with the following differences:

- `xvrf` interfaces have a fixed MAC address and cannot be configured
- `xvrf` interfaces are flagged NOARP, meaning that no ARP or NDP resolution is done when sending an IP packet through it
- `xvrf` interfaces do not support IP configuration

**See also:**

*veth interfaces.*

Here is an example of configuration where `xvrf` interfaces connect two VRF.

```

vrouters running config# / vrf vr1
vrouters running vrf vr1# interface xvrf to-vr2
vrouters running xvrf to-vr2#! link-interface to-vr1 link-vrf vr2
vrouters running xvrf to-vr2#! / vrf vr2
vrouters running vrf vr2#! interface xvrf to-vr1
vrouters running xvrf to-vr1#! link-interface to-vr2 link-vrf vr1
vrouters running xvrf to-vr1# commit

```

A YANG condition ensures that the binding of `xvrf` interfaces is consistent: the `xvrf` interfaces of a given pair must bind each other.

A route can then be added in `vr2` to reach a network 10.100.0.0/16 through `vr1`:

```

vrouters running config# vrf vr2
vrouters running vrf vr2# routing static

```

(continues on next page)

(continued from previous page)

```
vrouter running static# ipv4-route 10.100.0.0/16 next-hop to-vr1
vrouter running static# commit
```

Let's fetch the xvrf state inside vr1 after committing this configuration:

```
vrouter running config# show state vrf vr1 interface xvrf
xvrf to-vr2
  mtu 1500
  promiscuous false
  enabled true
  oper-status UP
  counters
    in-octets 360
    in-unicast-pkts 4
    in-discards 0
    in-errors 0
    out-octets 360
    out-unicast-pkts 4
    out-discards 0
    out-errors 0
    ..
  link-interface to-vr1
  link-vrf vr2
  ..
```

The same configuration can be made using this NETCONF XML configuration.

```
vrouter running config# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>vr1</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <xvrf xmlns="urn:6wind:vrouter/xvrf">
        <name>to-vr2</name>
        <enabled>true</enabled>
        <link-interface>to-vr1</link-interface>
        <link-vrf>vr2</link-vrf>
        (...)
      </xvrf>
    </interface>
  </vrf>
  <vrf>
    <name>vr2</name>
    <interface xmlns="urn:6wind:vrouter/interface">
```

(continues on next page)

(continued from previous page)

```

<xvrf xmlns="urn:6wind:vrouter/xvrf">
  <name>to-vr1</name>
  <enabled>true</enabled>
  <link-interface>to-vr2</link-interface>
  <link-vrf>vr1</link-vrf>
  (...)
</xvrf>
</interface>
</vrf>
</config>

```

**See also:**

The *command reference* for details.

An example of application of Cross-VRF interfaces is to provide vrf route leaking mechanisms with BGP. Cross-VRF interfaces are used to carry traffic from one VR to an other one. In the L<sub>3</sub>VPN case, the Cross-VRF interfaces can be the border between overlay and underlay information, as encapsulation and decapsulation operations will take place at this point.

**See also:**

The *BGP L3VPN* for details.

**Interface management****MAC**

The MAC address can be changed on ethernet interfaces.

To configure the MAC address of the existing interface eth0 in vrf main, do:

```

vrouter running config# vrf main
vrouter running vrf main# interface physical eth0
vrouter running physical eth0# ethernet mac-address 00:01:02:03:04:05
vrouter running physical eth0# commit

```

To display an interface MAC address:

```

vrouter> show state / vrf main interface physical eth0
physical eth0
  ipv6
    address fe80::dced:1ff:fec4:3a04/64
    ..
  mtu 2000
  port pci-b0s4

```

(continues on next page)

(continued from previous page)

```
counters
  in-octets 7316
  out-unicast-pkts 7
  out-octets 7316
  in-unicast-pkts 113
  in-discards 0
  in-errors 0
  out-discards 0
  out-errors 0
  ..
ethernet
  mac-address 00:01:02:03:04:05
  ..
oper-status UP
enabled true
..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter> show config xml absolute vrf main interface physical eth0
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <physical>
        <name>eth0</name>
        (...)
        <ethernet>
          <mac-address>00:01:02:03:04:05</mac-address>
        </ethernet>
      </physical>
    </interface>
  </vrf>
</config>
```

**See also:**

The *command reference* for details.

## MTU

Default MTU (Maximum Transmission Unit) interface is typically 1500. User can lower the value to cope with tunneling, or increase the value up to 9K to leverage jumbo support on the NIC.

To configure the MTU of the existing interface `eth0` in vrf `main` to 2000, do:

```
vrouter running config# vrf main
vrouter running vrf main# interface physical eth0
vrouter running physical eth0# mtu 2000
vrouter running physical eth0# commit
```

To display an interface mtu:

```
vrouter> show state / vrf main interface physical eth0
physical eth0
  ipv6
    address fe80::dced:1ff:fec4:3a04/64
    ..
  mtu 2000
  port pci-b0s4
  counters
    in-octets 7316
    out-unicast-pkts 7
    out-octets 7316
    in-unicast-pkts 113
    in-discards 0
    in-errors 0
    out-discards 0
    out-errors 0
    ..
  ethernet
    mac-address de:ed:01:c4:3a:04
    ..
  oper-status UP
  enabled true
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter> show config xml absolute vrf main interface physical eth0
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <physical>
```

(continues on next page)

(continued from previous page)

```
<name>eth0</name>
<mtu>2000</mtu>
(...)
</physical>
</interface>
</vrf>
</config>
```

**See also:**

The *command reference* for details.

## Physical Link Parameters

Unlike the *MAC* and *MTU* parameters which are applicable to all Ethernet interfaces, the following settings are only applicable to **physical** interfaces (i.e., interfaces backed by a physical PCI port).

**See also:**

The *command reference* for details.

## Auto-Negotiation

Even though it is often left enabled, auto-negotiation may be changed to cope with certain physical connections.

```
vrouter running physical eth0# ethernet auto-negotiation false
```

## Duplex Mode

By default, this setting is negotiated automatically with the connected endpoint. When auto-negotiation is set to false, you **must** specify the duplex mode of the connection.

```
vrouter running physical eth0# ethernet duplex-mode full|half
```

## Port Speed

By default, this setting is negotiated automatically with the connected endpoint. When auto-negotiation is set to false, you **must** specify the port speed.

```
vrouter running physical eth0# ethernet port-speed 10gb
```

## Flow Control

Pause frames are used by the NIC to ask a peer to slow down. It can be configured to automatically send pause frames as soon as the receive buffer is getting low, and to accept or not pause frames from other devices.

The default settings vary with hardware and device drivers. You may force them with the following commands:

```
vrouter running physical eth0# ethernet flow-control-tx false
vrouter running physical eth0# ethernet flow-control-rx false
```

## Statistics

Statistics about received and transmitted packets are available per interface.

To get the statistics of the eth0 interface in main vrf, do:

```
vrouter> show state vrf main interface physical eth0 counters
counters
  in-octets 7316
  out-unicast-pkts 22
  out-octets 7316
  in-unicast-pkts 113
  in-discards 0
  in-errors 0
  out-discards 0
  out-errors 0
```

To show the statistics in a human readable way:

```
vrouter running config# show interface statistics name eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode_
↳DEFAULT group default qlen 1000
  link/ether de:ad:de:01:02:03 brd ff:ff:ff:ff:ff:ff
  RX: bytes  packets  errors  dropped  overrun  mcast
  7316      113        0       0        0        0
  TX: bytes  packets  errors  dropped  carrier  collsns
  7316      22        0       0        0        0
```

You can also display the input/output packet and bit rate per second:

```
vrouter> show interface throughput name eth0
IFNAME          IN pkt/s  (IN bit/s)  OUT pkt/s  (OUT bit/s)
eth0             12.8M     (8.2G)      13.1M     (8.4G)
eth0             13.1M     (8.4G)      12.8M     (8.2G)
eth0             13.0M     (8.3G)      12.4M     (8.0G)
```

(continues on next page)

(continued from previous page)

|      |       |        |       |        |
|------|-------|--------|-------|--------|
| eth0 | 12.4M | (8.0G) | 13.0M | (8.3G) |
| ^C   |       |        |       |        |

The command can be interrupted by hitting `ctrl-c`.

**See also:**

The *command reference* for details about the API, and the *show interface* and *show interface throughput* commands.

### 3.1.6 IP Networking

#### IP

In this section we describe the IP configuration:

- *Static IP address*
- *DHCP for IPv4*
- *Static ARP/NDP neighbour entry*

IP configuration is available regardless the interface is either physical or virtual.

#### Static IP address

IPv4 or IPv6 address can be added to an interface. Let's add a static IPv4 address '10.0.0.1/24' on port we name 'giga0':

```
vrouter running config# vrf main
vrouter running vrf main# interface physical giga0
vrouter running physical giga0#! port pci-b0s4
vrouter running physical giga0# ipv4 address 10.0.0.1/24
```

Or an IPv6 address:

```
vrouter running physical giga0# ipv6 address 2001:DB8:657:494E::4401/64
```

Check the NETCONF XML for this configuration:

```
vrouter running physical giga0# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
```

(continues on next page)

(continued from previous page)

```

<physical>
  <name>giga0</name>
  <enabled>>true</enabled>
  <ipv4>
    <enabled>>true</enabled>
    <address>
      <ip>10.0.0.1/24</ip>
    </address>
  </ipv4>
  <ipv6>
    <router-advertisement>
      <suppress>>false</suppress>
    </router-advertisement>
    <enabled>>true</enabled>
    <dup-addr-detect-transmits>1</dup-addr-detect-transmits>
    <address>
      <ip>2001:DB8:657:494E::4401/64</ip>
    </address>
  </ipv6>
  <port>pci-b0s4</port>
</physical>
</interface>
</vrf>
</config>

```

To show the interface in a human readable way:

```

vrouter running config# show interface details name giga0
2: giga0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group_
↪default qlen 1000
link/ether de:ad:de:01:02:03 brd ff:ff:ff:ff:ff:ff
inet 10.0.0.1/24 brd 10.0.0.255 scope global giga0
    valid_lft forever preferred_lft forever
inet6 fec0::dcad:deff:fe01:203/64 scope site mngtmpaddr dynamic
    valid_lft 84443sec preferred_lft 12443sec
inet6 fe80::dcad:deff:fe01:203/64 scope link
    valid_lft forever preferred_lft forever
inet6 2001:db8:657:494e::4401/64 scope link
    valid_lft forever preferred_lft forever

```

## DHCP for IPv4

You can use the DHCP client to dynamically obtain an IP address and other parameters such as the default gateway, DNS servers information from a DHCP server. This parameter is not available for point to point interfaces.

In this example we enable DHCP on an interface, leaving only the following options activated:

- domain-name, used when resolving hostnames with DNS
- ntp-servers, to get the list of NTP servers
- interface-mtu, to get the MTU to use on this interface

```
vrouter running config# vrf main
vrouter running vrf main# interface physical eth0
vrouter running physical eth0#! port pci-b0s3
vrouter running physical eth0# ipv4 dhcp
vrouter running dhcp# del request subnet-mask
vrouter running dhcp# del request broadcast-address
vrouter running dhcp# del request time-offset
vrouter running dhcp# del request routers
vrouter running dhcp# del request domain-search
vrouter running dhcp# del request domain-name-servers
vrouter running dhcp# del request host-name
vrouter running dhcp# del request nis-domain
vrouter running dhcp# del request nis-servers
vrouter running dhcp# commit
```

To check the state:

```
vrouter running config# show state vrf main interface physical eth0 ipv4 dhcp
dhcp
  dhcp-lease-time 7200
  select-timeout 0
  current-lease
    expire 4 2018/06/28 16:14:53
    fixed-address 10.0.2.15
    rebind 4 2018/06/28 13:14:53
    renew 4 2018/06/28 02:49:27
  ..
  retry 300
  reboot 10
  enabled true
  initial-interval 10
  timeout 60
  request domain-name
  request ntp-servers
```

(continues on next page)

(continued from previous page)

```
request interface-mtu
..
```

Check the NETCONF XML for this configuration:

```
vrouter running physical eth0# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <physical>
        <name>eth0</name>
        <enabled>true</enabled>
        <ipv4>
          <enabled>true</enabled>
          <dhcp>
            <enabled>true</enabled>
            <timeout>60</timeout>
            <retry>300</retry>
            <select-timeout>0</select-timeout>
            <reboot>10</reboot>
            <initial-interval>10</initial-interval>
            <dhcp-lease-time>7200</dhcp-lease-time>
            <request>domain-name</request>
            <request>ntp-servers</request>
            <request>interface-mtu</request>
          </dhcp>
        </ipv4>
        <ipv6>
          <router-advertisement>
            <suppress>>false</suppress>
          </router-advertisement>
          <enabled>true</enabled>
          <dup-addr-detect-transmits>1</dup-addr-detect-transmits>
        </ipv6>
        <port>pci-b0s3</port>
      </physical>
    </interface>
  </vrf>
</config>
```

## Static ARP/NDP neighbour entry

Static ARP (IPv4) or NDP (IPv6) neighbour can be added to an interface. This parameter is not available for point to point interfaces.

From 'ipv4' context you can add static ARP entries to bind an IP address to a fixed ethernet address:

```
vrouter running ipv4# neighbor 10.0.0.64 link-layer-address 00:06:57:49:4e:44
```

Or from 'ipv6' context, you can add static NDP entries to bind an IPv6 address to a fixed ethernet address:

```
vrouter running ipv6# neighbor 2001:DB8:657:494E::4499 link-layer-address_
↪00:06:57:49:4e:44
```

Check the NETCONF XML for this configuration:

```
vrouter running physical eth0# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <physical>
        <name>eth0</name>
        <enabled>true</enabled>
        <ipv4>
          <enabled>true</enabled>
          <neighbor>
            <ip>10.0.0.64</ip>
            <link-layer-address>00:06:57:49:4e:44</link-layer-address>
          </neighbor>
        </ipv4>
        <ipv6>
          <enabled>true</enabled>
          <neighbor>
            <ip>2001:DB8:657:494E::4499</ip>
            <link-layer-address>00:06:57:49:4e:44</link-layer-address>
          </neighbor>
        </ipv6>
      </physical>
    </interface>
  </vrf>
</config>
```

---

**Note:** CG-NAT requires a Turbo CG-NAT Application License.

---

## CG-NAT

CG-NAT, also known as Large Scale NAT (LSN), is an extension of NAT for large scale networks and ISPs (Internet Service Providers).

The key advantages of CG-NAT compared to NAT are:

- **High Transparency:** CG-NAT implements multiple advanced features like Endpoint-Independent Mapping, Endpoint-Independent Filtering, address pooling and port parity preservation. These features provide better experience to ‘nated’ users and allow scaling.
- **Fairness and Resource Sharing:** CG-NAT provides options to limit the number of connections per user. This ensures that resources are equitably shared between the different users.
- **Optimized Logging system:** CG-NAT can generate large amounts of logging data. CG-NAT implements a feature called port block allocation to limit the number of log entries by grouping per port range.
- **Transition between IPv6-only and IPv4-only networks** thanks to NAT64, in conjunction to DNS64.

**Caution:** Stateful IP packet filtering and CG-NAT are exclusive. If CG-NAT is enabled, stateful IP packet filtering must be disabled on ports bound to the fast path.

### See also:

The *CG-NAT command reference* and the *CG-NAT fast path limits command reference* for details.

- *Configuration*
  - *Pool*
  - *Rule*
  - *Understanding Port Block Allocation*
  - *Address Pooling*
  - *Port algorithm*
  - *Active Block Timeout*
  - *Endpoint mapping*
  - *Endpoint filtering*
  - *Hairpinning*
  - *Conntracks*
  - *ALG*
  - *Summary*
- *Logging*

- *dynamic*
- *deterministic*
- *Troubleshooting*
  - *Invalid packet state statistics*
  - *State/NAT/USER/Block Allocation Failures*
  - *No IP Public errors*
  - *NAT port allocation failures*
  - *Maximum number of blocks reached*
  - *Full IP Public errors*
- *Dimensioning*

## Configuration

### Pool

A CG-NAT pool contains a list of IPv4 addresses used to change the IPv4 or IPv6 source address and port of a packet.

The CG-NAT implements a feature called port block allocation. Each time a new user sent a packet through the CG-NAT router, a block of ports (i.e. a port range) from one of the IPv4 addresses in the pool is allocated to this one. The number of ports given to an user is configurable via the `block-size` option.

Here is an example of a CG-NAT pool:

```
vrouter running config# vrf main
vrouter running vrf main# cg-nat
vrouter running cg-nat!# pool mypool
vrouter running mypool!# address 192.0.2.33-192.0.2.49
vrouter running mypool!# address 192.0.1.0/24
vrouter running mypool!# address 192.0.3.1
vrouter running mypool!# block-size 512
```

A pool needs to be associated to a CG-NAT rule, see the next section.

## Rule

We have three types of NAT rules:

- dynamic: Private addresses are dynamically mapped to public addresses: for each private address, a public address is dynamically allocated from a pool and associated to the private address.
- deterministic: each private address is always mapped to the same public address and is assigned a dedicated port pool at the time of configuration
- static: Mapping between private addresses and public ones are static. Public addresses are configured directly in the rule.

### static SNAT44

All packets routed through an output interface and matching the filtering criteria defined in a CG-NAT rule are source nated with an ip.

Here is an example of a CG-NAT rule:

```
vrouter running mypool!# ..
vrouter running cg-nat#! rule 1
vrouter running rule 1#! static-snat44
vrouter running static-snat44#! match
vrouter running match#! source ipv4-address 100.64.0.0/32
vrouter running match#! outbound-interface eth1
vrouter running static-snat44#! translate-to
vrouter running translate-to#! ipv4-address 192.0.2.1
vrouter running translate-to# commit
```

To display the applied configuration:

```
vrouter running config# show state vrf main cg-nat
cg-nat
  enabled true
  rule 1
    static-snat44
      match
        source
          ipv4-address 100.64.0.1/32
          ..
        outbound-interface eth1
        ..
      translate-to
        ipv4-address 192.0.2.1
        ..
```

(continues on next page)

(continued from previous page)

```

    ..
    ..
logging
    enabled false
    ..
..

```

## static DNAT44

All packets received on an input interface and matching the filtering criteria defined in a CG-NAT rule are destination nated with an ip.

Here is an example of a CG-NAT rule:

```

vrouters running mypool!# ..
vrouters running cg-nat#! rule 1
vrouters running rule 1#! static-dnat44
vrouters running static-dnat44#! match
vrouters running match#! destination ipv4-address 192.0.2.1/32
vrouters running match#! inbound-interface eth1
vrouters running static-dnat44#! translate-to
vrouters running translate-to#! ipv4-address 100.64.0.1
vrouters running translate-to# commit

```

To display the applied configuration:

```

vrouters running config# show state vrf main cg-nat
cg-nat
    enabled true
    rule 1
        static-dnat44
            match
                destination
                    ipv4-address 192.0.2.1/32
                ..
            inbound-interface eth1
            ..
        translate-to
            ipv4-address 100.64.0.1
            ..
        ..
    ..
logging

```

(continues on next page)

(continued from previous page)

```

    enabled false
    ..
    ..

```

## dynamic SNAT44

All packets routed through an output interface and matching the filtering criteria defined in a CG-NAT rule are source nated with an ip dynamically assigned from one CG-NAT pool.

Here is an example of a CG-NAT rule:

```

vrouter running mypool!# ..
vrouter running cg-nat#! rule 1
vrouter running rule 1#! dynamic-snat44
vrouter running dynamic-snat44#! match
vrouter running match#! source ipv4-address 100.64.0.0/10
vrouter running match#! outbound-interface eth1
vrouter running dynamic-snat44#! translate-to
vrouter running translate-to#! pool-name mypool
vrouter running translate-to# commit

```

To display the applied configuration:

```

vrouter running config# show state vrf main cg-nat
cg-nat
  enabled true
  pool mypool
    address 192.0.1.1-192.0.1.254
    address 192.0.2.33-192.0.2.49
    address 192.0.3.1
    block-allocation-mode dynamic
    block-size 512
    port-range
      1024
      65535
    ..
  ..
  rule 1
    dynamic-snat44
      match
        source
          ipv4-address 100.64.0.0/10
        ..

```

(continues on next page)

(continued from previous page)

```

        outbound-interface eth1
        ..
    translate-to
        pool-name mypool
        max-contracks-per-user 0
        max-blocks-per-user 1
        active-block-timeout 0
        user-timeout 120
        port-algo parity
        endpoint-mapping independent
        endpoint-filtering independent
        hairpinning false
        address-pooling paired
        ..
    ..
..
logging
    enabled false
    ..
..

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouter running config# show config xml absolute vrf main cg-nat
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <cg-nat xmlns="urn:6wind:vrouter/cgnat">
      <enabled>true</enabled>
      <logging/>
      <pool>
        <name>mypool</name>
        <port-range/>
        <address>192.0.2.33-192.0.2.49</address>
        <address>192.0.1.0/24</address>
        <address>192.0.3.1</address>
        <block-allocation-mode>dynamic</block-allocation-mode/>
        <block-size>512</block-size>
      </pool>
      <rule>
        <id>1</id>
        <dynamic-snat44>
          <match>
            <source>

```

(continues on next page)

(continued from previous page)

```

        <ipv4-address>100.64.0.0/10</address>
    </source>
    <outbound-interface>eth1</outbound-interface>
</match>
<translate-to>
    <pool-name>mypool</pool-name>
</translate-to>
</dynamic-snat44>
</rule>
</cg-nat>
</vrf>
</config>

```

## deterministic SNAT44

The deterministic-snat44 works like dynamic-snat44, except that IP and ports from the CG-NAT pool are associated in a deterministic manner to the user at the time of configuration: the CG-NAT pool is partitioned in chunk of port block, so that each user has its own port block. See the Understanding Port Block Allocation/deterministic for more information about this point.

Here is an example of a deterministic CG-NAT rule:

```

vrouter running mypool!# del block-size
vrouter running mypool!# block-allocation-mode deterministic
vrouter running mypool!# ..
vrouter running cg-nat#! rule 1
vrouter running rule 1#! deterministic-snat44
vrouter running deterministic-snat44#! match
vrouter running match#! source ipv4-address 100.64.0.0/10
vrouter running match#! outbound-interface eth1
vrouter running deterministic-snat44#! translate-to
vrouter running translate-to#! pool-name mypool
vrouter running translate-to# commit

```

## dynamic SNAT64

In the case of NAT64, translation between IPv6-only addresses on the private side and IPv4 addresses on the public side is taken care of by mapping the IPv4 public addresses to a special IPv6 prefix that is configured in the translate-to section of the rule.

DNS64 must be configured to translate IPv4 addresses DNS replies to IPv6 ones using this prefix.

The rest of the configuration is similar to dynamic NAT44.

```

vrouter running config# vrf main
vrouter running vrf main# cg-nat
vrouter running cg-nat#! rule 1
vrouter running rule 1#! dynamic-snat64
vrouter running dynamic-snat64#! match
vrouter running match#! source ipv6-address fd00:100::/64
vrouter running match#! outbound-interface eth2
vrouter running match#! ..
vrouter running dynamic-snat64#! translate-to
vrouter running translate-to#! pool-name mypool
vrouter running translate-to#! max-blocks-per-user 4
vrouter running translate-to#! destination-prefix 64:ff9b::/96
vrouter running translate-to# .. . . . .
vrouter running vrf main# dns
vrouter running dns# proxy
vrouter running proxy# dns64 64:ff9b::/96
vrouter running dns64 64:ff9b::/96# client fd00:100::/64
vrouter running dns64 64:ff9b::/96# exclude 64:ff9b::/96
vrouter running dns64 64:ff9b::/96# mapped not 10.0.0.0/8 192.168.0.0/16 172.16.0.0/12

```

The `client` option can be used to select which client the DNS64 function will apply to. The CPE (Customer Premise Equipment) subnet in the CG-NAT rule should have the same value.

The `exclude` option is a safeguard to ensure that no IPv6 address matching `64:ff9b::/96` will be returned to IPv6 clients. Without it the CG-NAT might translate IPv6 packets going to this IPv6 server into IPv4 packet with an incorrect IPv4 address.

The `mapped` option can be used to avoid mapping specific IPv4 address to IPv6. For example, it is a good idea not to embed any RFC 1918 addresses that name servers on the Internet might inadvertently return.

## static SNAT64

All IPv6 packets routed through an output interface and matching the filtering criteria defined in a CG-NAT rule are source nated with an IPv4 address.

Here is an example of a CG-NAT rule:

```

vrouter running mypool!# ..
vrouter running cg-nat#! rule 1
vrouter running rule 1#! static-snat64
vrouter running static-snat64#! match
vrouter running match#! source ipv6-address fd00:100::1/128
vrouter running match#! outbound-interface eth1
vrouter running static-snat64#! translate-to
vrouter running translate-to#! ipv4-address 192.0.2.1

```

(continues on next page)

(continued from previous page)

```
vrouter running translate-to# commit
```

## static DNAT46

All IPv4 packets received on an input interface and matching the filtering criteria defined in a CG-NAT rule are destination nated with an IPv6 address.

Here is an example of a CG-NAT rule:

```
vrouter running mypool!# ..
vrouter running cg-nat#! rule 1
vrouter running rule 1#! static-dnat64
vrouter running static-dnat64#! match
vrouter running match#! destination ipv6-address 192.0.2.1/32
vrouter running match#! inbound-interface eth1
vrouter running static-dnat64#! translate-to
vrouter running translate-to#! ipv4-address fd00:100::1/128
vrouter running translate-to# commit
```

## Understanding Port Block Allocation

### dynamic

It is a legal requirement for an ISP to be able to provide the mapping between a user and a public IP address at a given point in time. With classic NAT, it means that each new user session has to be logged. This is obviously not scalable. Additionally with classic NAT, a user can consume all the ports of the public IP.

To reduce the amount of logs and equitably share the ports of the different public IPs, CG-NAT provides the Port Block Allocation (PBA) feature that consists in allocating blocks of ports to each user. As logging is done per block of ports, the amount of logs is reduced. And as the number and size of the blocks can be configured, the user port consumption is controlled. Here is how PBA works.

Each time a new user sends a packet through the vRouter, a block of ports is allocated to him from one of the IP addresses in the pool. The public IPs are selected using a round-robin algorithm. Each public IP is divided into blocks of ports, whose size and range is defined in the pool configuration. This prevents a single user from consuming all ports.

Here is an example to change the number of ports per block:

```
vrouter running cg-nat# / vrf main cg-nat pool mypool block-size 256
vrouter running cg-nat# commit
```

For the next session of the same user, a port is allocated from its block of ports, until the block is exhausted. In that case, a new block can be allocated for this user and it becomes the active block. There is only one active block per user. The maximum number of blocks per user is defined in the rule configuration.

Here is an example to change the maximum number of blocks per user:

```
vrouter running cg-nat# / vrf main cg-nat rule 1 translate-to
max-blocks-per-user 2
vrouter running cg-nat# commit
```

Since ports are allocated from the same block (until this one is empty), port prediction can potentially happen. To randomize port allocation, it is possible to allocate a new active block if the current active block has been active for too long, even if there are still some ports available in the active block. This feature is called active block timeout. As it can increase the average numbers of blocks allocated per user, it is disabled by default.

Here is an example to configure an active block timeout of 60 seconds:

```
vrouter running cg-nat# / vrf main cg-nat rule 1 translate-to
active-block-timeout 60
vrouter running cg-nat# commit
```

When all the ports are released from a non-active block, this one is released immediately. Regarding the active block, the block is only released when the user subscription times out. The default user timeout is 120 seconds.

Here is an example to change the user timeout:

```
vrouter running cg-nat# / vrf main cg-nat rule 1 translate-to user-timeout 180
vrouter running cg-nat# commit
```

## deterministic

ISPs have a legal requirement to be able to map a subscriber's inside address with the address and port used on the public Internet (e.g., for abuse response). With dynamic allocation, any new block of port allocated/released need to be logged.

The port allocation algorithm for deterministic is predictable and sequential (i.e. the first block goes to address 1, the second block to address 2, etc.).

Thanks to this allocation method, it is possible to retrieve at any time a mapping between a public IP and a private one without any log. The following commands can be used for this purpose:

```
vrouter> show cg-nat deterministic-public-block user-address 10.100.0.59
10.175.0.3 15046-15300
vrouter> show cg-nat deterministic-user-address public-address 10.175.0.3 public-port_
↪17001
10.100.0.66
vrouter> show cg-nat deterministic-mappings | pager
```

(continues on next page)

(continued from previous page)

```
10.100.0.0: 10.175.0.3 1-255
10.100.0.1: 10.175.0.3 256-510
10.100.0.2: 10.175.0.3 511-765
...
```

A deterministic rule needs to be associated to a pool with block-allocation-mode set to deterministic. Public IPs cannot be shared between deterministic and dynamic rules.

For deterministic rules, the block-size option is not mandatory. It is even recommended to not use it. Because it is automatically computed to this maximal value with the following formula:  $(\text{port\_range\_of\_the\_pool} * \text{number\_pool\_ips}) / \text{number\_of\_user}$ .

The users always have one block per protocol. As a consequence, deterministic rules don't support the 'max-blocks-per-user' option.

## Address Pooling

The address-pooling option defines if a public IP address is paired to a user:

- paired (default value): It means that the same public IP address is used for all sessions originating from the same user.
- no-paired: It means that different public IP addresses can be used for different sessions originating from the same user.

It's recommended to use paired address pooling for applications that require all sessions associated with one private IP address to be translated to the same public IP address for multiple sessions.

## Port algorithm

The port-algo defines which method is used to allocate ports:

- parity (default): This algorithm preserves the parity of the port, i.e. an even port will be mapped to an even port, and an odd port will be mapped to an odd port.
- random: This algorithm chooses a port randomly.

Here is an example to configure random port allocation.

```
vrouter running translate-to# port-algo random
vrouter running translate-to# commit
```

### Active Block Timeout

The active-block-timeout defines how the active block changes:

- 0 (default): The active block changes only on port allocation, if it is full.
- any other value: The active block also changes everytime the timeout expires.

---

**Note:** As this option can increase the average number of blocks allocated per user, it is disabled by default.

---

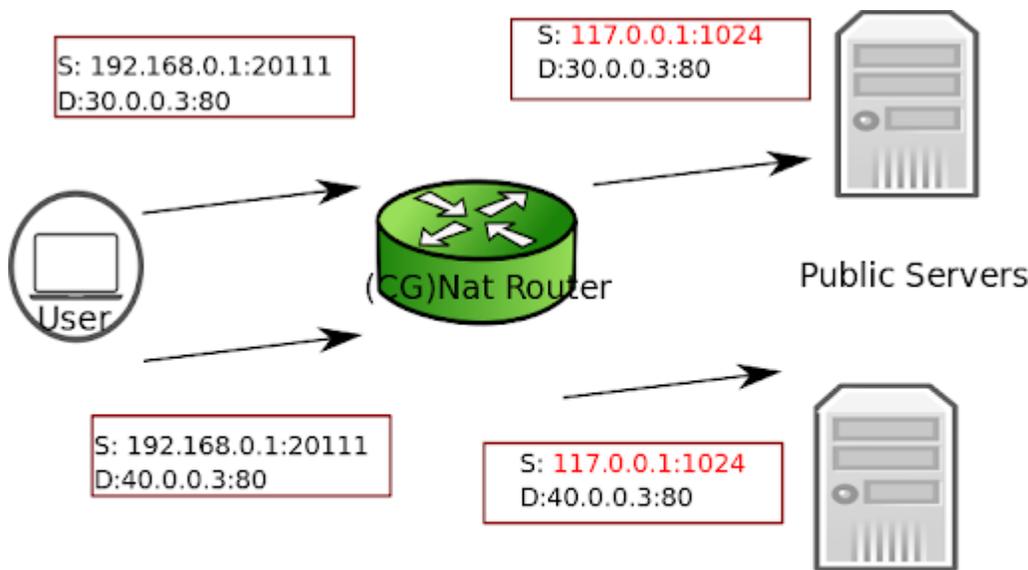
Here is an example to configure active block timeout.

```
vrouter running translate-to# active-block-timeout 60
vrouter running translate-to# commit
```

### Endpoint mapping

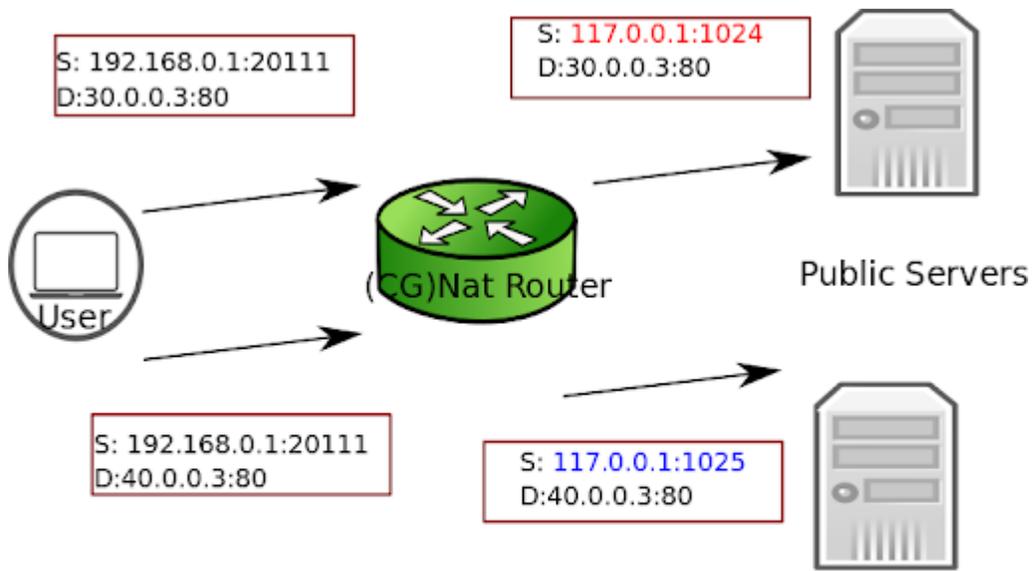
There are two endpoint mapping behaviors:

- independent (default): The vRouter reuses the same port mapping for subsequent packets sent from the same internal IP address and port to any external IP address and port.



independent endpoint-mapping  
same mapping for different sessions

- dependent: The vRouter reuses the same port mapping for subsequent packets sent from the same internal IP address and port to the same external IP address and port.



dependent endpoint-mapping  
a mapping only applies to one session

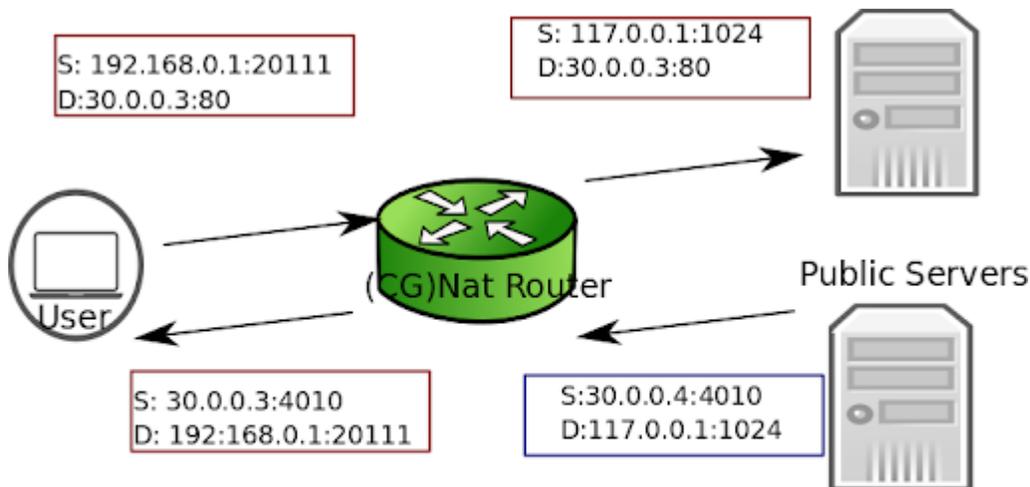
Here is an example to configure dependent endpoint mapping.

```
vrouter running translate-to# endpoint-mapping dependent
vrouter running translate-to# commit
```

## Endpoint filtering

There are two endpoint filtering behaviors:

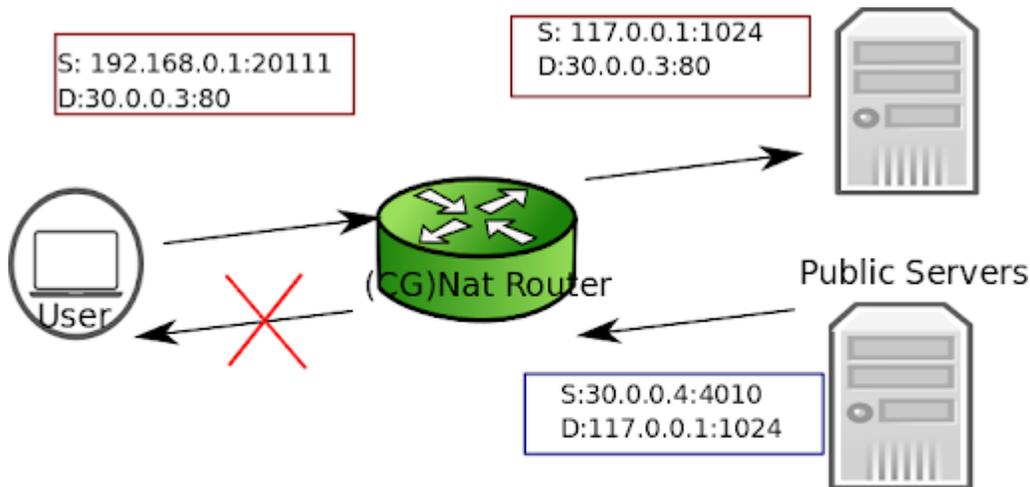
- independent (default): Inbound packets from external endpoints are only filtered out if their destination IP address and port don't match an existing public IP address and port mapping.



independent endpoint-filtering

a mapping can be reused by any external session

- dependent: Inbound packets from external endpoints are filtered out if they don't match an existing mapping (source and destination IPs and ports, protocol).



dependent endpoint-filtering

a mapping can be reused by only one external session

Here is an example to configure dependent endpoint filtering.

```
vrouter running translate-to# endpoint-filtering dependent
vrouter running translate-to# commit
```

## Hairpinning

The hairpinning feature allows two endpoints (user 1 and user 2) on the private network to communicate together using their public IP addresses and ports.

By default, the hairpinning feature is disabled. To enable it, use the following command.

```
vrouter running translate-to# hairpinning true
vrouter running translate-to# commit
```

## Contracks

It is possible to set contrack timeouts for each protocol. UDP, ICMP and GRE protocols only handle basic contrack states (new, established, closed), whereas TCP offers more granularity.

```
vrouter running config# vrf main cg-nat contrack timeouts tcp
syn-sent          simsyn-sent      syn-received     established      fin-sent
fin-received      closed           close-wait       fin-wait         last-ack
time-wait

vrouter running config# vrf main cg-nat contrack timeouts udp
new                established      closed
```

It is also possible to change TCP behavior for specific cases.

```
vrouter running config# vrf main cg-nat contrack behavior
tcp-window-check      tcp-rst-strict-order
```

## ALG

Application-level gateways allow to use specific applications through CG-NAT.

```
vrouter running config# vrf main cg-nat alg
ftp          h323-q931   h323-ras    pptp          rtsp          sip-tcp
sip-udp      tftp        dns-udp
```

### Summary

The following table summarizes the different parameters for CG-NAT configuration.

**Warning:** Changing some configuration parameters requires to flush users. This is automatically done when required (see the table below), and causes a service interruption.

| Cat-<br>e-<br>gory | Parame-<br>ter               | Flush | Note  |
|--------------------|------------------------------|-------|---|
| pool               | address                      | no    | New addresses can be added without any impact. If an address is removed, all users assigned to it are flushed.  |
|                    | block-size                   | yes   |   |
|                    | port-range                   | yes   |   |
| rule               | match                        | no    | Contracks not matching the new criteria are not flushed. Contracks matching the new criteria are flushed.   |
|                    | pool-name                    | yes   |   |
|                    | max-<br>blocks-per-<br>user  | no    | Extra blocks are not flushed for users having more blocks than the new max. They become inactive and will be flushed after all sessions are released. |
|                    | active-<br>block-<br>timeout | no    | The new timeout starts after the current one expires.   |
|                    | user-<br>timeout             | no    | The new timeout starts after the current one expires.   |
|                    | port-algo                    | no    | For new blocks only.  |
|                    | endpoint-<br>mapping         | no    | For new mappings only.  |
|                    | endpoint-<br>filtering       | no    | For new mappings only.  |
|                    | hairpin-<br>ning             | no    | For new mappings only.  |
|                    | address-<br>pooling          | no    | For new users only.   |

## Logging

### dynamic

You can enable logs for CG-NAT to track each port block allocation/deallocation for a user :

```
vrouter running config# vrf main cg-nat logging enabled true
commit
```

The logs can be displayed with the following command:

```
vrouter running config# show log service cgnat
Jun 11 08:02:46 vrouter systemd[1]: Started Fast Path cgnat log daemon.
Jun 11 08:02:46 vrouter fp-cgnat-logd[4269]: CGNAT Log listen on 5001
Jun 11 08:03:09 vrouter fp-cgnat-logd[4269]: USER 100.64.0.1 (matching rule 1): NEW_
↳BLOCK (pool "mypool", ip public 192.0.2.33, proto 6, port 1 - 512) at Tue Jun 11_
↳08:03:09 2019
Jun 11 08:07:30 vrouter fp-cgnat-logd[4269]: USER 100.64.0.1 (matching rule 1):_
↳DESTROY BLOCK (pool "mypool", ip public 192.0.2.33, proto 6, port 1 - 512) at Tue_
↳Jun 11 08:07:30 2019
```

The following fields are displayed for each port block allocation/deallocation:

- IP address of the user
- name of the CG-NAT rule matched by the user
- type of event: NEW BLOCK (a new port range is associated to this user) or DESTROY BLOCK (port range is released for this user).
- pool and ip used to nat the user
- port range of the block
- l4 protocol (i.e. 6 for tcp, 17 for udp, 1 for icmp, 47 for gre)
- timestamp of the event

### deterministic

The deterministic algorithm is predictable, so block allocation/deallocations don't need to be logged. Anyway, to retrieve the mapping between user and public address, the deterministic algorithm needs to know the deterministic CG-NAT is configured.

In consequence, the deterministic CG-NAT configuration is logged.

```
vrouter running config# show log service cgnat
Mar 02 14:32:06 dut-vm systemd[1]: Started Fast Path cgnat log daemon.
```

(continues on next page)

(continued from previous page)

```

Mar 02 14:32:06 dut-vm fp-cgnat-logd[3526]: CGNAT Log listen on 5001
Mar 02 14:32:06 dut-vm fp-cgnat-logd[3526]: cgnat-deterministic-conf;none at 2021-03-
↪02T14:32:06Z
Mar 02 14:35:42 dut-vm fp-cgnat-logd[3526]: cgnat-deterministic-conf;
↪mypool|sequential1.0|10.100.0.0/24|10.175.0.3|255|1-65535 at 2021-03-02T14:35:42Z

```

The deterministic CG-NAT configuration can be used to retrieve a user address behind a public address/port with the fp-cgnat-deterministic application.

```

# fp-cgnat-deterministic get-cpe-ip 10.175.0.3 17001 \
'cgnat-deterministic-conf;mypool|sequential1.0|10.100.0.0/24|10.175.0.3|255|1-65535'
10.100.0.66

```

The 'show cg-nat deterministic-public-block/deterministic-user-address/deterministic-mappings' commands can also be used to retrieve a mapping for a previous deterministic CG-NAT configuration by using the date-and-time parameter. This one is able to retrieve the configuration applied at this time and get the information requested with fp-cgnat-deterministic application.

```

vrouter> show cg-nat deterministic-user-address public-address 10.175.0.3 public-port-
↪17001 date-and-time '2021-03-03 11:21:00'
10.100.0.66

```

## Troubleshooting

### Invalid packet state statistics

To display the CG-NAT statistics, use the following command.

```

vrouter> show cg-nat statistics
...
Invalid packet state cases:
...
 0 TCP case RST
...
 0 TCP case I
 0 TCP case II
 0 TCP case III
...

```

If the TCP case I, II or III statistics are incremented, you may disable TCP window checks as follows.

```

vrouter> edit running
vrouter running config# vrf main cg-nat contrack

```

(continues on next page)

(continued from previous page)

```
vrouter running contrack# behavior tcp-window-check enabled false
vrouter running contrack# commit
```

If the TCP case RST statistic is incremented, you may disable TCP RST strict ordering as follows.

```
vrouter> edit running
vrouter running config# vrf main cg-nat contrack
vrouter running contrack# behavior tcp-rst-strict-order enabled false
vrouter running contrack# commit
```

---

**Note:** Disabling these features improves performance to the detriment of TCP robustness.

---

### State/NAT/USER/Block Allocation Failures

```
vrouter> show cg-nat statistics
...
State and NAT entries:
...
  0 state allocation failures
...
  0 NAT entry allocation failures
  0 NAT port allocation failures
CGNat entries:
...
  0 USER allocation failures
...
  0 Block allocation failures
...
```

If one of these statistics is incremented, it means that one of the memory pools of the vRouter is full. Memory pool usage can be dumped using the following command.

```
vrouter> show cg-nat mempool-usage
cgnat_user_pool : 2000/10000 (20.00%)
cgnat_block_pool : 8000/80000 (10.00%)
table_pool : 0/1056 (0.00%)
conn_pool : 1056736/1056736 (100.00%)
nat_pool : 1056736/1056736 (100.00%)
```

In the example above, the connection and NAT memory pools are full. Their size must be increased as follows.

```
vrouters running config# / system
fast-path limits cg-nat
vrouters running cg-nat# max-contracks 2000000
vrouters running cg-nat# max-nat-entries 2000000
vrouters running cg-nat# commit
```

Refer to the capability tuning section.

### No IP Public errors

```
vrouters> show cg-nat statistics
...
CGNat entries:
...
  0 No IP Public
...
```

If this statistic is incremented, it means there are no blocks available in any public IP. This can be checked using the following command.

```
vrouters> show cg-nat pool-usage pool-name mypool
tcp block usage: 4095/4095 (100.0%)
udp block usage: 4095/4095 (100.0%)
icmp block usage: 4095/4095 (100.0%)
gre block usage: 4095/4095 (100.0%)
```

To solve this issue, add a new public IP to the pool using the following command.

```
vrouters> edit running
vrouters running config# vrf main cg-nat pool mypool
vrouters running pool mypool# address 32.96.120.0/24
vrouters running pool mypool# commit
```

### NAT port allocation failures

There are two main reasons for port allocation failures:

- A user has consumed all its port blocks. The maximum number of blocks per user can be increased in the rule using the max-blocks-per-user command.
- No blocks are available on the public IP allocated to the user. In this case, the Full IP Public statistic is also incremented.

To list users with allocation failures to understand how many users are impacted, use the following command.

```
vrouters> show cg-nat user rule-id 1 threshold-errors 1
100.64.0.1 -> 32.96.119.108
  2/2 tcp blocks, 0/2 udp blocks, 0/2 icmp blocks, 0/2 gre blocks
  63 no port errors, 0 no block errors, 0 full public ip errors
```

To understand why a specific user has many connections, use the following command.

```
vrouters> show cg-nat contracks rule-id 1 user-address 100.64.0.1
CON:
  vrfid 0 flags 0x6 alg none tsdiff 47 timeout 120
  forw proto 6 100.64.0.1:1024-> 32.96.118.2:6001 hash:be3505a5
  back proto 6 32.96.118.2:6001-> 32.96.119.108:1216 hash:92e65736
  state 10:
    F { end 0 maxend 0 mwin 0 wscale 0 flags 1}
    T { end 0 maxend 0 mwin 0 wscale 0 flags 0}
  NAT: original address 100.64.0.1 proto 6 oport 1024 tport 1216
CON:
  vrfid 0 flags 0x6 alg none tsdiff 56 timeout 120
  forw proto 6 100.64.0.1:65024-> 32.96.118.2:6000 hash:913f8bf7
  back proto 6 32.96.118.2:6000-> 32.96.119.108:1024 hash:27051895
  state 10:
    F {end 0 maxend 0 mwin 0 wscale 0 flags 1}
    T {end 0 maxend 0 mwin 0 wscale 0 flags 0}
  NAT: original address 100.64.0.1 proto 6 oport 65024 tport 1024
...

```

## Maximum number of blocks reached

If the maximum number of blocks is reached, it probably means that you have not allocated enough blocks per user. You can collect some statistics to get average/percentile block and port usage of all users with the following commands.

```
vrouters> show cg-nat block-statistics rule-id 1
block-usage:
  1 user (with > 1 block = 1, ratio 100.00%)
  blocks per user: min = 2, max = 2, average = 2.00
  1 user (100.00%) have 2 blocks
vrouters> show cg-nat port-statistics rule-id 1
port-usage:
  1 user (with > 1 port = 1, ratio 100.00%)
  ports per user: min = 128, max = 128, average = 128.00
  1 user (100.00%) have 128 ports
```

You can also check the block usage and port usage of a specific user to get more details with the following commands.

```
vrouter> show cg-nat blocks rule-id 1 user-address 100.64.0.1
BLOCK: status active, proto 1 tports 1024 - 1031 parity=1, usage 1/8
vrouter> show cg-nat port-usage rule-id 1 user-address 100.64.0.1
tcp session usage: 0/16
udp session usage: 0/16
icmp session usage: 1/16
gre session usage: 0/16
```

Then, you can decide to increase the number of blocks per user or the block size. Refer to the Changing parameters section.

### Full IP Public errors

```
vrouter> show cg-nat statistics
...
CGNat entries:
...
  0 Full IP Public
...
```

The paired address pooling feature ensures the assignment of the same public IP address to all sessions originating from the same internal user, as described in RFC 4787 Req 2 (<https://tools.ietf.org/html/rfc4787#page-22>).

It means that when a user has started to use one public IP address, all its port blocks will be allocated from this same IP. Adding a new public IP to the pool won't solve the issue, as the user cannot allocate a block from a new public IP.

A possible way to recover such situation is to add new IP address to the pool, and then flush all the current connections of all users, as follows.

```
vrouter running config# / vrf main cg-nat pool mypool
vrouter running pool mypool# address 32.96.120.0/24
vrouter running pool mypool# commit
Configuration committed.

vrouter running pool mypool# flush cg-nat user rule-id 1
```

## Dimensioning

The maximum numbers for NAT entries, CPEs (users), conntracks (sessions), blocks and block sizes are defined in the configuration. These limits can be adjusted to adapt to the amount of memory available in the system.

```
vrouter running config# system fast-path limits cg-nat
max-conntracks      max-nat-entries  max-users          max-blocks
max-block-size
```

The following table shows a list of different limit combinations and the corresponding memory requirement. This is empirical and may have to be tuned according to your use case.

| Max conntracks | Max nat entries | Max cpe | Max blocks | Required memory |
|----------------|-----------------|---------|------------|-----------------|
| 1M             | 1M              | 10K     | 80K        | 5 GB            |
| 2M             | 2M              | 20K     | 80K        | 6 GB            |
| 4M             | 4M              | 20K     | 80K        | 8 GB            |
| 8M             | 8M              | 20K     | 80K        | 12 GB           |
| 16M            | 16M             | 20K     | 80K        | 24 GB           |
| 30M            | 30M             | 20K     | 80K        | 32 GB           |

**Warning:** Changing these values triggers a restart of the fast-path (hence, a service interruption). Therefore you should carefully think about your dimensioning before launching your Turbo Router into production.

Modifying these limits will automatically restart the fast path and interrupt packet processing. To check that the fast path is back up and running, use the following command.

```
vrouter running config# show state system fast-path
fast-path
  enabled stopping
  ..
vrouter running config# show state system fast-path
fast-path
  enabled starting
  ..
vrouter running config# # show state system fast-path
fast-path
  enabled true
  ...
```

## NAT

NAT provides a way to translate IPv4 addresses and ports while crossing the device. This technique is typically used to conserve addresses now that IPv4 addresses become a scarce resource.

---

**Note:** NAT rules not configured by the management system will not be displayed by show state and will be lost when a new NAT configuration is committed.

---

**Caution:** NAT and CG-NAT are exclusive. If CG-NAT is enabled, NAT must be disabled on ports bound to the fast path.

### See also:

The *command reference* for details.

- *Relation with IP packet filtering*
- *Source NAT*
- *Masquerading*
- *Destination NAT*

## Relation with IP packet filtering

When configuring NAT along with IP packet filtering, you should know that:

- source NAT happens in postrouting chain, after mangle table
- destination NAT happens in prerouting chain, after raw and mangle tables
- connection tracking (conntracks) keeps track of how the packet should be changed in the two directions

### See also:

*IP packet filtering* for details.

## Source NAT

Source NAT changes the source IPv4 address or port of an outgoing packet.

---

**Note:** A destination NAT rule is not needed to do source NAT. Connection tracking keeps track of how the packet should be changed in the two directions, so a source NAT rule is enough. A destination NAT rule can be added if the NAT connection can be opened from both sides.

---

Here is an example of a rule which matches the packets with source address 1.1.1.1 output to public interface, and translates their source address to 2.2.2.2:

```
vrouter running config# vrf main
vrouter running vrf main# nat
vrouter running nat# source-rule 1 source address 1.1.1.1/32 outbound-interface public
↳translate-to address 2.2.2.2
vrouter running nat# commit
```

To display the applied configuration:

```
vrouter running config# show state vrf main nat
nat
  source-rule 1 source address 1.1.1.1/32 outbound-interface public translate-to
↳address 2.2.2.2
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running config# show config xml absolute vrf main nat
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <nat xmlns="urn:6wind:vrouter/nat">
      <source-rule>
        <id>1</id>
        <source>
          <address>
            <value>1.1.1.1/32</value>
          </address>
        </source>
        <outbound-interface>
          <name>public</name>
        </outbound-interface>
        <translate-to>
          <address>
```

(continues on next page)

(continued from previous page)

```

        <value>2.2.2.2</value>
      </address>
    </translate-to>
  </source-rule>
</nat>
</vrf>
</config>

```

## Masquerading

Masquerading is a kind of source NAT. It is a way to use one public IPv4 address visible on the Internet for an entire private network, using the IPv4 address of the device public interface.

Here is an example of a rule which matches the packets sent via public interface, and translates their source address to the IPv4 address of public interface:

```

vrouters running config# vrf main
vrouters running vrf main# nat
vrouters running nat# source-rule 1 outbound-interface public translate-to output-
↪address
vrouters running nat# commit

```

To display the applied configuration:

```

vrouters running config# show state vrf main nat
nat
  source-rule 1 outbound-interface public translate-to output-address
  ..

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouters running config# show config xml absolute vrf main nat
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <nat xmlns="urn:6wind:vrouter/nat">
      <source-rule>
        <id>1</id>
        <outbound-interface>
          <name>public</name>
        </outbound-interface>
        <translate-to>
          <output-address/>

```

(continues on next page)

(continued from previous page)

```

    </translate-to>
  </source-rule>
</nat>
</vrf>
</config>

```

## Destination NAT

Destination NAT changes the destination IPv4 address or port of an incoming packet.

---

**Note:** A source NAT rule is not needed to do destination NAT. Connection tracking keeps track of how the packet should be changed in the two directions, so a destination NAT rule is enough. A source NAT rule can be added if the NAT connection can be opened from both sides.

---

Here is an example of a rule which matches the tcp packets with destination port 8080 received from public interface, and translates their destination address to 2.2.2.2, and their destination port to 80:

```

vrouter running config# vrf main
vrouter running vrf main# nat
vrouter running nat# destination-rule 1 protocol tcp destination port 8080 inbound-
↳interface public translate-to address 2.2.2.2 port 80
vrouter running nat# commit

```

To display the applied configuration:

```

vrouter running config# show state vrf main nat
nat
  destination-rule 1 protocol tcp destination port 8080 inbound-interface public_
↳translate-to address 2.2.2.2 port 80
  ..

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouter running config# show config xml absolute vrf main nat
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <nat xmlns="urn:6wind:vrouter/nat">
      <destination-rule>
        <id>1</id>
        <protocol>
          <value>tcp</value>

```

(continues on next page)

(continued from previous page)

```
</protocol>
<destination>
  <port>
    <value>8080</value>
  </port>
</destination>
<inbound-interface>
  <name>public</name>
</inbound-interface>
<translate-to>
  <address>
    <value>2.2.2.2</value>
    <port>80</port>
  </address>
</translate-to>
</destination-rule>
</nat>
</vrf>
</config>
```

## 3.1.7 Routing

### Static routes

#### Standard routing

Once the IP addresses have been configured, communication is possible between the nodes (hosts or routers) directly connected to the same IP sub-network. It is a one hop communication. To communicate with other nodes that are connected to a different sub-network, a dedicated node, the router, requires routes. For example, you can define static IP routes to link sub-networks.

Static routes do not scale and are not error-free. They should be used only when dynamic routing protocols cannot be deployed, or in case of very simple topologies.

You can implement static routing by directly manipulating the equipment routing table. It may be used with any dynamic routing protocol. When both static and dynamic routes are set, the static ones are preferred because their administrative distance is 1.

To add a static route, do:

```
vrouter running config# vrf main
vrouter running vrf main# routing static
vrouter running static# ipv4-route 10.200.0.0/24 next-hop 10.125.0.2
```

(continues on next page)

(continued from previous page)

```
vrouter running static# commit
Configuration applied.
```

To display the static routes state:

```
vrouter running config# show state vrf main routing static
static
  ipv4-route 10.200.0.0/24
    next-hop 10.125.0.2
  ..
..
```

To check the route is present in the system Routing Information Base:

```
vrouter running config# show state vrf main routing rib
rib
  ipv4-route 10.200.0.0/24
    next-hop 10.125.0.2
      selected true
      distance 1
      protocol static
      uptime 00:11:55
      interface ntfp2
      active true
      fib true
    ..
  [...]
..
..
```

To show the state in a human readable way:

```
vrouter running config# show ipv4-routes vrf main
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

K>* 0.0.0.0/0 [0/0] via 10.0.2.2, mgmt0, 00:02:00
C>* 10.0.2.0/24 is directly connected, mgmt0, 00:02:00
C>* 10.100.0.0/24 is directly connected, ntfp1, 00:02:00
C>* 10.125.0.0/24 is directly connected, ntfp2, 00:02:00
C>* 10.175.0.0/24 is directly connected, ntfp3, 00:02:00
S>* 10.200.0.0/24 [1/0] via 10.125.0.2, ntfp2, 00:02:00
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running config# show config xml absolute vrf main routing static
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <routing xmlns="urn:6wind:vrouter/routing">
      <static>
        <ipv4-route>
          <destination>10.200.0.0/24</destination>
          <next-hop>
            <next-hop>10.125.0.2</next-hop>
          </next-hop>
        </ipv4-route>
      </static>
    </routing>
  </vrf>
</config>
```

#### See also:

- The *command-reference* for details.

## Policy-Based Routing

Turbo Router supports multiple routing tables. By default, routes are set in the main table as explained above. For PBR, it is possible to specify the table to use using an identifier. See the *Policy-Based Routing* section for details.

To add a static route in a specific table, do:

```
vrouter running static# table 100 ipv4-route 10.200.0.0/24 next-hop 10.175.0.2
vrouter running static# commit
Configuration applied.
```

To show the state for a specific table:

```
vrouter running config# show config xml absolute vrf main routing static
vrouter running config# show ipv4-routes vrf main table 100
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

S>* 10.200.0.0/24 [1/0] via 10.175.0.2, ntfp3, 00:02:00
```

The matching NETCONF XML is as follows:

```
vrouter running config# show config xml absolute vrf main routing static
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <routing xmlns="urn:6wind:vrouter/routing">
      <static>
        <table>
          <table-id>100</table-id>
          <ipv4-route>
            <destination>10.200.0.0/24</destination>
            <next-hop>
              <next-hop>10.175.0.2</next-hop>
            </next-hop>
          </ipv4-route>
        </table>
      </static>
    </routing>
  </vrf>
</config>
```

## BFD With Static Routes

It is possible to enable a failover mechanism that relies on nexthops configured in static routes. By monitoring with BFD that nexthop, the route will be either validated or invalidated, according to BFD status. This mechanism enforces the reachability. To get more information on BFD, please see *BFD*.

Below example illustrates how a BFD peer session context is created, associated to next-hop 10.125.0.2.

```
vrf customer1
  routing static ipv4-route 10.200.0.0/24 next-hop 10.125.0.2 track bfd
```

Then you can continue the configuration as usual. For timer settings, the default emission and reception settings are set to 300000 microseconds, which may not be what is wished. In that case, it is possible to override default timers, by configuring general timer settings. More information is given in *Configuring general BFD settings*.

## BFD Configuration And Monitoring In Static Routing Using Trackers

It's also possible to configure a BFD or ICMP tracker manually. This enables using the same tracker in different services. An example is available in the *BGP configuration and monitoring using trackers* documentation.

More information about tracker can be found in *Tracker guide*.

## Routing utilities

Routing packets requires to handle the core element of a routing table : the prefix. Prefix is generally an IPv4 or an IPv6 address associated with a mask. There are needs on routing protocols to have tools that permit apply some filtering. This is true for BGP, but it is also true for OSPF. Some information is given about 2 useful tools that are used on the above mentioned routing protocols : IPv4 Access-Lists and IPv4 Prefix-List.

Also, this chapter presents the route-map object. This objects works on the match/set mechanism. It is feeded by input given by routing protocols, and it returns an output that is modified to be conform with the set rules contained in the route-map.

Finally, this chapter gives an overview about routing priorities between the various routing protocols, by explaining the distance.

- *IPv4/IPv6 Access-Lists*
- *IP Prefix List*
- *Route-Maps*
- *Routing Administrative Distance*
- *Logging*

## IPv4/IPv6 Access-Lists

Configure the IPv4 access-list

```
vrouter running config# routing
vrouter running routing# ipv4-access-list ACCESS-LIST-NAME {permit|deny} A.B.C.D/M.
↳[exact-match]
vrouter running routing# commit
```

It is possible to give a description to an access list by typing the command

```
vrouter running routing# ipv4-access-list ACCESS-LIST-NAME remark "comment between.
↳inverted commas"
```

As described, a prefix will match an access-list entry if that prefix is included in that access-list entry. It is possible to override the behaviour with the `exact-match` keyword so that the access-list will need to match the exact prefix value.

Conversely, it is possible to create IPv6 Access List:

```
vrouter running config# routing
vrouter running routing# ipv6-access-list ACCESS-LIST-NAME {permit|deny} X:X::X:X/M.
↳[exact-match]
```

(continues on next page)

(continued from previous page)

```
vrouter running routing# ipv6-access-list ACCESS-LIST-NAME remark "comment between
↳inverted commas"
vrouter running routing# commit
```

The below prefix-list should be preferred to the access-lists described here.

## IP Prefix List

A prefix filter is more powerful than an access-list filter to process the network prefixes.

In comparison to access-list prefix-list have the following advantages:

- Can process a range of values
- Performance improvement in prefix lookup of large lists
- More flexible

Filtering by prefix list involves the following rules :

- An empty prefix list permits all prefixes.
- An implicit deny is assumed if a given prefix does not match any entries of a prefix list.
- When multiple entries of a prefix list match a given prefix, the longest match is chosen.
- The router prefix-list lookup begins at the top with sequence number 1, if a match occurs then the router do not go through the rest of the prefix list.

The syntax to define a prefix filter is:

```
vrouter running config# routing
vrouter running routing# ipv4-prefix-list PREFIX-LIST-NAME
vrouter running ipv4-prefix-list# seq SEQ policy {permit|deny} [address PREFIX/M
    [prefix-min A | prefix-max B]]
vrouter running routing# ipv6-prefix-list PREFIX-LIST-NAME
vrouter running ipv6-prefix-list# seq SEQ policy {permit|deny} [address PREFIX/M
    [prefix-min A | prefix-max B]]
```

**PREFIX-LIST-NAME** unique identifier name of the prefix list context

**SEQ** Sequence of the rule named PREFIX-LIST-NAME Range varies from 1 to 4294967295

**PREFIX/M** Network prefix and M the length of the mask. The format is an IPv4 address for an IPv4 prefix list, or an IPv6 address for an IPv6 prefix list.

**A and B** A and B range goes from 0 to 32 for an IPv4 prefix list, while it goes from 0 to 128 for an IPv6 prefix list. Those integers up to 32 that can be used to form a block of prefixes. A, B and M are such as:

$$M < A$$

$M < B$

$A ( B$

$M < A ( B ( 32$

### Example with IPv4 prefix list

Let  $P1/m$  be a network prefix that matches  $PREFIX/M$ . For example  $PREFIX/M$  could be  $192.168.0.0/16$  and  $P1/m$  could be  $192.168.10.0/24$ .

Moreover, if  $A$  and  $B$  are defined,  $P1/M$  matches this rule if  $M$  is greater or equal than  $A$  and if  $M$  is less or equal to  $B$  ( $A ( M ( B$ ). For example  $192.168.10.0/24$  matches the rule 5, however it does not match the rule 10.

```
vrouter running routing# ipv4-prefix-list PREFIX-FILTER-NAME
vrouter running ipv4-prefix-list# seq 5 policy permit address 192.168.0.0/16 prefix-
←min 17 prefix-max 25
vrouter running ipv4-prefix-list#
```

The prefix lists can be used in many cases:

```
route-map:
    match ip address prefix-list FILTER-NAME
    match ipv6 address prefix-list FILTER-NAME
    match ip next-hop address prefix-list FILTER-NAME

neighbor configuration:
    neighbor A.B.C.D address-family ADDRESSFAMILY
        prefix-list {in|out} prefix-list-name FILTER-NAME
```

#### Note:

- The command ‘match ip/ipv6 address’ can be used with an access-list too. However, you can check that the syntax is not exactly the same: `match ip address prefix-list FILTER-NAME` vs. `match ip address access-list ACCESS-LIST-NAME`.

### Route-Maps

Route-Maps operate on the match/set mechanism. it applies a set of actions to the incoming entries that matches the set criteria. Incoming entries stand for routing information. For instance, BGP updates.

To create a route-map object, use the following command:

```
vrouter running routing# route-map ROUTEMAP-NAME seq SEQ policy {permit|deny}
vrouter running route-map SEQ#
```

**ROTEMAP-NAME** unique identifier name of the route-map context

**SEQ** Sequence of the rule named ROUTEMAP-NAME. Range varies from 1 to 65535

The route-map introduces a sequence number that permits introducing several match/set rules sequentially. If the first sequence does not match the incoming entry, then the next sequence is looked up.

The match and set operations vary from one routing protocol to another one. BGP gathers a wide variety of match/set combinations. Here below is depicted some basic examples:

To configure a route-map based on a peer criterion, and apply a weight to the routing entry, use the following command:

```
vrouters running routing# route-map ROUTEMAP-NAME seq SEQ policy {permit|deny}
vrouters running route-map SEQ# match peer A.B.C.D
vrouters running route-map SEQ# set weight (0-4294967295)
```

---

**Note:** Some route-map actions and/or match conditions can be protocol-specific (for instance matching on community-id makes sense only for BGP). If the associated protocol is not configured and activated, the specific items will not be displayed in a `show state` but will still be visible on a `show configuration`.

---

## Routing Administrative Distance

Actually, even if prefixes can be filtered, the origin of the route entry is kept, and a weight is associated to each route entry, according to the origin of the routing protocol. That weight is called the administrative distance. For instance, if the same prefix has 2 entries in both static routing table, and BGP routing table, the prefix with the least administrative distance will be chosen locally and installed in the system. Here it will be the static routing table.

We give here a reminder of the common routing protocols administrative distance:

| Routing protocol            | Administrative distance |
|-----------------------------|-------------------------|
| Connected prefixes (routes) | 0                       |
| Static routes               | 1                       |
| iBGP (Internal BGP)         | 200                     |
| eBGP (External BGP)         | 20                      |
| OSPF v2 and OSPF v3         | 110                     |
| RIP and RIPv2               | 120                     |

## Logging

Routing logging options are configurable from the global routing context:

```
vrouter running config# routing logging
```

All logs are sent to the daemon syslog facility. By default, only messages of severity higher than **error** are logged. This can be modified by changing the **level** option:

```
vrouter running logging# level LEVEL
```

**LEVEL** Severity from which messages should be logged.

Here is the list of severities from the most serious to the least:

| severity  | description                        |
|-----------|------------------------------------|
| emergency | System is unusable.                |
| alert     | Action must be taken immediately.  |
| critical  | Critical conditions.               |
| error     | Error conditions.                  |
| info      | Informational messages.            |
| notice    | Normal but significant conditions. |
| warning   | Warning conditions.                |
| debug     | Debug-level messages.              |

The verbosity of these logs is configurable per routing protocol. See the *routing global command reference guide* for details.

## BGP

### BGP Overview

As the Internet is composed of many ASes (Autonomous Systems): ISPs, universities, multi-homed networks... the inter-AS (Autonomous System) routing policies are getting more and more complex. BGP is the today's EGP (External Gateway Protocol), which handles these policies between the ASes. The border gateway is the router that interconnects many ASes. BGP allows you to create loop-free interdomain routing between ASes.

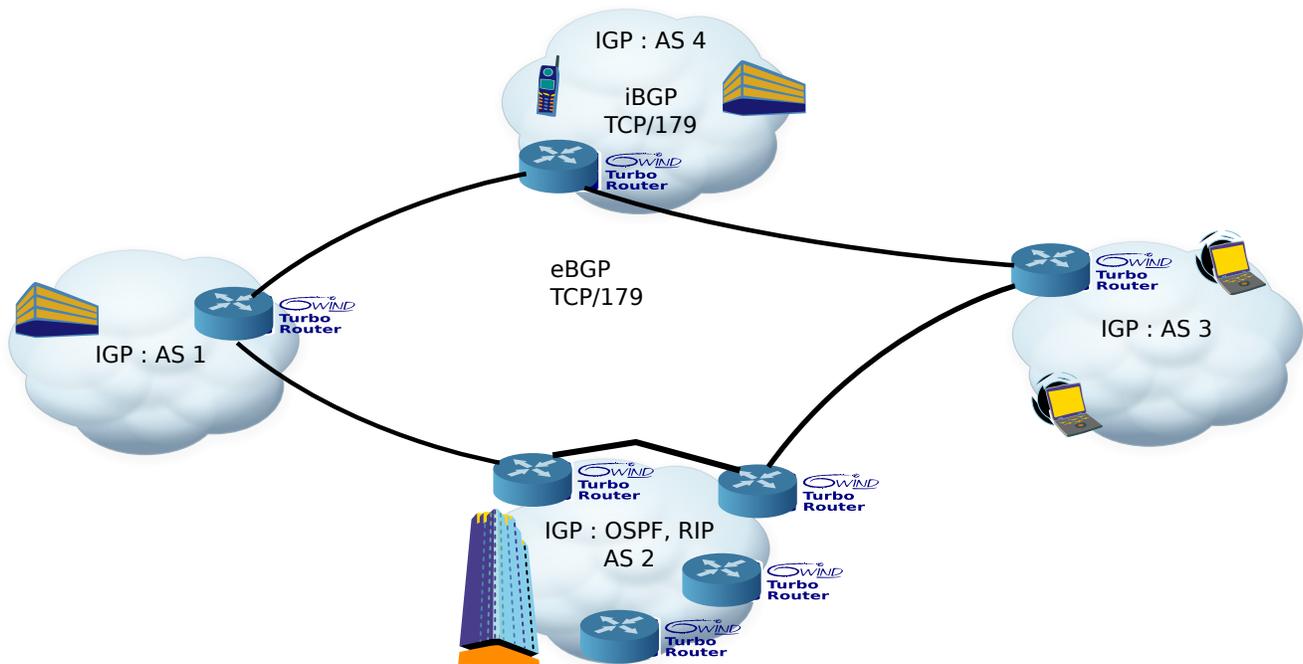


Fig. 2: The EGP (BGP) vs. the IGPs (Internal Gateway Protocols) (RIP, RIPNG, OSPF v2, OSPF v3)

BGP came up at the beginning of the 90's. The first RFC of BGP was published in 1989. The today's BGP is referred to as BGP 4, which was described by the **RFC 1771** (<https://tools.ietf.org/html/rfc1771.html>). It is an exterior routing protocol that distributes some network reachability information. These network information are a set of network prefixes, which could be either IPv4 or IPv6 network prefixes; and the reachability information are a list of ASNs (Autonomous System Numbers) that are crossed to reach some network prefixes.

BGP runs over the unicast TCP on the well-known port 179. The same TCP port is used for both IPv4 and IPv6.

Any two routers which have established a TCP connection to exchange BGP routing information are called BGP peers or BGP neighbors. The two peers begin by exchanging their full BGP table, then incremental updates are sent when the routing tables change.

When BGP is running between two routers belonging to different ASes it is called Exterior BGP, sometimes referenced as eBGP. When the two routers belong to the same AS it is called interior BGP, referenced as iBGP.

In routing protocols design rules, the BGP routing protocol is mainly used to exchange routing information between different Autonomous Systems. Within the same AS, routing information is exchanged through an IGP (Internal Gateway Protocol) routing protocol like RIP or OSPF.

Today, BGP is used to establish peering for various usages:

- Transit peering: Forwarding data to/from other ISP's networks.
- Downstream peering: Forwarding data to/from enterprises.
- Private peering: Establish VPN connectivity for customers with multiple sites, for instance. L3VPN and L2VPN (Layer 2 Virtual Private Network) are some use cases.

- Public peering with IXP (Internal Exchange Point) to benefit from numerous partners.

The BGP is handled by FRR (<https://frrouting.org/>).

Turbo Router provides the following features:

**RFC 1771** (<https://tools.ietf.org/html/rfc1771.html>): A Border Gateway Protocol 4 (BGP-4 (Border Gateway Protocol 4))

**RFC 1997** (<https://tools.ietf.org/html/rfc1997.html>): BGP Communities Attribute

**RFC 2545** (<https://tools.ietf.org/html/rfc2545.html>): Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing

**RFC 4364** (<https://tools.ietf.org/html/rfc4364.html>): BGP/MPLS IP Virtual Private Networks (VPNS)

**RFC 2796** (<https://tools.ietf.org/html/rfc2796.html>): BGP Route Reflection An alternative to full mesh iBGP

**RFC 2858** (<https://tools.ietf.org/html/rfc2858.html>): Multiprotocol Extensions for BGP-4

**RFC 3065** (<https://tools.ietf.org/html/rfc3065.html>): AS confederations for BGP

**RFC 4360** (<https://tools.ietf.org/html/rfc4360.html>): BGP Extended Communities Attribute

**RFC 4456** (<https://tools.ietf.org/html/rfc4456.html>): BGP Route Reflection: An Alternative to Full Mesh Internal BGP

**RFC 4384** (<https://tools.ietf.org/html/rfc4384.html>): bgp/mpls IP Virtual Private Networks (VPNs)

**RFC 5575** (<https://tools.ietf.org/html/rfc5575.html>): Dissimination of Flow Specification Rules

**RFC 6793** (<https://tools.ietf.org/html/rfc6793.html>): BGP support for Four-Octet Autonomous System (AS) Number Space

**RFC 7911** (<https://tools.ietf.org/html/rfc7911.html>): Advertisement of Multiple Paths in BGP

**RFC 8093** (<https://tools.ietf.org/html/rfc8093.html>): BGP Large Communities Attribute

**RFC 8212** (<https://tools.ietf.org/html/rfc8212.html>): Default External BGP Route Propagation Behavior without Policies

**RFC 7432** (<https://tools.ietf.org/html/rfc7432.html>): BGP Large Communities Attribute

**RFC 8365** (<https://tools.ietf.org/html/rfc8365.html>): A Network Virtualization Overlay solution Using Ethernet VPN (EVPN)

**RFC 6480** (<https://tools.ietf.org/html/rfc6480.html>): An Infrastructure to Support Secure Internet Routing

**RFC 8210** (<https://tools.ietf.org/html/rfc8210.html>): The Resource Public Key Infrastructure to Router Protocol, Version 1

Draft EVPN Inter Subnet Forwarding (<https://tools.ietf.org/html/draft-ietf-bess-evpn-inter-subnet-forwarding-08>)

**See also:**

The *command reference* for details.

## BGP configuration

There are a list of necessary elements to know when forging a BGP configuration.

- *Basic elements for configuration*
- *Basic BGP configuration*
- *Peer-groups*
- *Route-Reflector*
- *Multipath*

### Basic elements for configuration

When forging a BGP configuration, the local AS number, and the remote AS number, as well as the remote IP address have to be known in order to establish peering.

An AS is an administrative set of routers, depending on an administrative authority. There are public or assigned ASes, and private ASes. An ASes is identified by a number called ASN (Autonomous System Number).

The public ASNs are any registered ASN values that are not private. These ASNs are assigned by a RIR (Regional Internet Registry). The private ASNs are made up of 2 ranges that can be used for local administration. These numbers are 64512 through 65535, and 4200000000 through 4294967294.

BGP has been extended to exchange routing information not only for IPv4 routing tables, also other routing information like IPv6, or for other purpose: flowspec, or L3VPN or L2VPN. The address-family command allows us to identify the network protocol. It is made up of a pair of arguments AFI, SAFI. For instance, by default, IPv4, unicast is enabled and stands for the routing information of IPv4.

Here below is an example on how to configure a sample BGP configuration with both IPv4 and IPv6 address-family set:

```
vrf main
  routing
    bgp
      router-id 10.125.0.1
      as 65501
      neighbor 10.125.0.3
        remote-as 65502
        address-family ipv6
      ..
    ..
  commit
```

The same configuration can be made using this NETCONF XML configuration:

```

vrouter running config# show config xml absolute vrf main routing bgp
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <routing xmlns="urn:6wind:vrouter/routing">
      <bgp xmlns="urn:6wind:vrouter/bgp">
        <router-id>10.125.0.1</router-id>
        <as>65501</as>
        <neighbor>
          <neighbor-address>10.125.0.3</neighbor-address>
          <address-family>
            <ipv6-unicast>
              </ipv6-unicast>
            </address-family>
            <remote-as>65502</remote-as>
          </neighbor>
        </bgp>
      </routing>
    </vrf>
  </config>

```

Configuring various address-family means that there are subtle differences between each address-family, that permit benefiting from each specificity.

For instance, IPv6, unicast address-family provides 2 IPv6 next-hops : the local one and the global one.

Also, IPv4, vpn is the L3VPN combination for MPLS tunnels. While the routing information exchanged deals with inner IPv4 information, the MPLS VPN (Virtual Private Network) SAFI (Subsequent Address-Family Identifier) implies that the overlay will be based with MPLS. The nexthop information will stand for underlay tunnel end point information. Here, the nexthop may be either IPv4 or IPv6, independently of the inner IPv4 prefix. The nexthop will also contain the MPLS label identifier.

---

**Note:** You can also disable BGP, either by suppressing the configuration:

---

```

vrf main
  del routing bgp
  ..

```

Alternatively, if you don't want to lose the configuration, and disabling BGP configuration, you can use following command:

```

vrf main
  routing bgp
    enabled false

```

This method can be used if the user wants to force peering with remote BGP speakers. Consecutively changing the state of BGP will force the peering. Here, below illustration indicates how the session for 10.125.0.3 is flushed.

```
flush bgp vrf main ipv4 unicast neighbor 10.125.0.3
```

Note that this command can also selectively flush different parts of the routing tables, like ADJ-RIB-IN (Adjacency RIB Inbound) by issuing the `soft in` prefix at the end of the command. An other possibility is to disable the whole BGP instance.

```
vrf main
  routing bgp enabled false
  commit
  routing bgp enabled true
  commit
```

### Basic BGP configuration

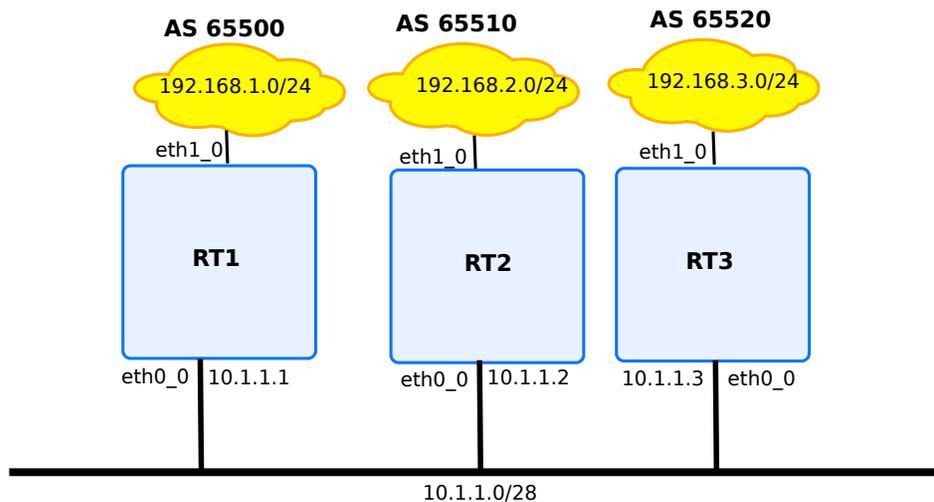


Fig. 3: BGP configuration illustration with 3 BGP peerings

The above diagram depicts 3 devices, each one has a BGP instance that peers with each other. The 3 devices configuration is like below:

**rt1**

```
routing bgp
  router-id 10.1.1.1
  as 65500
  neighbor 10.1.1.2 remote-as 65510
  neighbor 10.1.1.3 remote-as 65520
  address-family ipv4-unicast redistribute connected
  ..
  ..
interface
  physical eth1_0
    ipv4 address 192.168.1.0/24
    ..
  ..
  physical eth0_0
    ipv4 address 10.1.1.1/28
    ..
  ..
```

**rt2**

```
routing bgp
  router-id 10.1.1.2
  as 65510
  neighbor 10.1.1.1 remote-as 65500
  neighbor 10.1.1.3 remote-as 65520
  address-family ipv4-unicast redistribute connected
  ..
  ..
interface
  physical eth1_0
    ipv4 address 192.168.2.0/24
    ..
  ..
  physical eth0_0
    ipv4 address 10.1.1.2/28
    ..
  ..
```

**rt3**

```

routing bgp
  router-id 10.1.1.3
  as 65520
  neighbor 10.1.1.1 remote-as 65500
  neighbor 10.1.1.2 remote-as 65510
  address-family ipv4-unicast redistribute connected
  ..
  ..
interface
  physical eth1_0
    ipv4 address 192.168.3.0/24
    ..
  ..
  physical eth0_0
    ipv4 address 10.1.1.3/28
    ..
  ..

```

After having executed the three configurations, the status of the BGP connections can be obtained. The peerings between the devices can be visualised with the following command:

```
rt1> show bgp summary
```

```

IPv4 Unicast Summary:
BGP router identifier 10.1.1.1, local AS number 65500 vrf-id 0
BGP table version 5
RIB entries 9, using 1368 bytes of memory
Peers 2, using 41 KiB of memory

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ OutQ  Up/Down State/P
fxRcd
10.1.1.2      4      65510   17     17      0    0    0 00:09:08 4
10.1.1.3      4      65520   17     17      0    0    0 00:09:11 4

Total number of neighbors 2

```

The output of the state column must be blank in case the BGP connection is established, otherwise it reflects the state of the BGP connection. The different BGP session states are studied later in the section. Following command gives detailed BGP information about a given neighbor:

```

rt1> show bgp neighbor 10.1.1.2
BGP neighbor is 10.1.1.2, remote AS 65510, local AS 65500, external link
Hostname: rt1

```

(continues on next page)

(continued from previous page)

```

BGP version 4, remote router ID 10.1.1.2
BGP state = Established, up for 00:14:00
Last read 00:01:00, Last write 00:01:00
Hold time is 180, keepalive interval is 60 seconds
Neighbor capabilities:
  4 Byte AS: advertised and received
  AddPath:
    IPv4 Unicast: RX advertised IPv4 Unicast and received
  Route refresh: advertised and received(old & new)
  Address Family IPv4 Unicast: advertised and received
  Hostname Capability: advertised (name: rt1,domain name: n/a)
    received (name: rt2,domain name: n/a)
  Graceful Restart Capabilty: advertised and received
    Remote Restart timer is 120 seconds
  Address families by peer:
    none
  Graceful restart informations:
    End-of-RIB send: IPv4 Unicast
    End-of-RIB received: IPv4 Unicast
  Message statistics:
    Inq depth is 0
    Outq depth is 0

      Sent      Rcvd
  Opens                1        1
  Notifications:      0        0
  Updates:             6        6
  Keepalives:         14       14
  Route Refresh:      0        0
  Capability:         0        0
  Total:              21       21
    
```

It is also possible to dump the list of BGP entries that rt1 learnt from the other peers, by using following command on configuration mode:

```

rt1> show bgp ipv4 unicast neighbors
BGP table version is 5, local router ID is 10.1.1.1, vrf id 0
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
                i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

  Network          Next Hop          Metric LocPrf Weight Path
* 10.0.2.0/24      10.1.1.2          0           0 65510 ?
*                  10.1.1.3          0           0 65520 ?
    
```

(continues on next page)

(continued from previous page)

|                |          |   |                 |
|----------------|----------|---|-----------------|
| *>             | 0.0.0.0  | 0 | 32768 ?         |
| * 10.1.1.0/28  | 10.1.1.2 | 0 | 0 65510 ?       |
| *              | 10.1.1.3 | 0 | 0 65520 ?       |
| *>             | 0.0.0.0  | 0 | 32768 ?         |
| *> 192.168.1.0 | 0.0.0.0  | 0 | 32768 ?         |
| * 192.168.2.0  | 10.1.1.2 |   | 0 65520 65510 ? |
| *>             | 10.1.1.2 | 0 | 0 65510 ?       |
| * 192.168.3.0  | 10.1.1.3 |   | 0 65510 65520 ? |
| *>             | 10.1.1.3 | 0 | 0 65520 ?       |

Displayed 5 routes and 11 total paths

## Peer-groups

Scaling BGP deployments may be useful, when one deploys multiple instances of BGP. Instead of configuring each peer one by one, it is possible to configure peer groups.

A peer group is defined by a name, and is being applied the same configuration as the one applied to a single peer IP, except for the IP addressing of that peer.

You can use following configuration to create a peer group named group.

```

routing bgp
  listen
    neighbor-range 10.135.0.0/24 neighbor-group group
  ..
as 65502
neighbor-group group
  address-family
    ipv6-unicast
  ..
  ..
  remote-as 65501
  update-source 10.135.0.2
  ..
neighbor 10.145.0.2
  neighbor-group group
  ..
  ..

```

By default, the peer group will create as many peering connection as it receives incoming BGP connections that match its settings. It is however possible to limit the number of accepted incoming connections by establishing a range of potential IP addresses. Conversely, it is also possible to define some peers with outgoing peering, with the inherited configuration coming from the peer-group.

## Route-Reflector

Route reflector is used in iBGP networks, where the number of BGP peers becomes too important. Instead of using a full mesh peering, a 1-N peering topology is used. A single ( or two, in case backup is needed) BGP instance acts as route reflector server, and receives/replies BGP updates from/to route reflector clients accordingly. This permits scaling some setups. Creating a route reflector server consists in defining an IBGP peering session, either via peer-group or by defining directly a peer. The option `route-reflector-client` must be set to true.

```
as 65501
neighbor-group group
  address-family
    ipv4-unicast
      route-reflector-client true
    ..
  ..
  remote-as 65501
neighbor 1.1.1.1
  address-family
    ipv4-unicast
      route-reflector-client true
    ..
  ..
  remote-as 65501
  ..
```

There is no need to add extra-configuration to the iBGP clients.

## Multipath

BGP multipath permits to create ECMP routes, so that traffic can be load-shared on all the available routing entries. By default, BGP know how to handle up to 8 ECMP route entries. It is possible to reduce per the number of maximum-paths per address-family, and for both iBGP or eBGP sessions. Here is a configuration example, on how to disable multipath for IPv4, unicast BGP:

```
router-id 10.125.0.1
address-family
  ipv4-unicast
    maximum-path
      ebgp 1
      ibgp 1
    ..
  ..
as 65501
```

The multipath is criteria are strict by default. That means that even if as-path attribute that goes along with the prefixes differs, then the load-sharing will fail. There are some mitigations methods that permit relax the load sharing. For instance, the as-path attribute list can be completely ignored with following command, thus permitting to do load sharing across paths that do not share at all same path-list.

```
bestpath
  as-path
    ignore true
  ..
  ..
```

It is also possible to find an intermediary point, by taking into account only prefixes that share different path list, but same as-path list count.

```
bestpath
  as-path
    multipath-relax as-set
  ..
  ..
```

BGP supports advertisements of multiple paths. This is an extra identifier that is encoded in the NLRI (Network Layer Reachability Information) of the packet. It contains a separate identifier. For instance, it permits to transmit 2 ECMP entries that will be differentiated by that identifier. To enable encoding of the prefix with the add-path option, use the following configuration command:

```
neighbor 10.125.0.3
  remote-as 65502
  address-family
    ipv4-unicast
      addpath
        tx-all-paths true
    ..
  ..
  ..
  ..
```

## BGP configuration options

The BGP routing protocol is very rich and offers many options. In this paragraph we will study the most used and useful BGP options.

- *Aggregation*
  - *No aggregation flags*

- *Summary-only aggregation flag*
- *As-set aggregation flag*
- *Combined summary-only and as-set aggregation flags*
- *Confederation*
- *Overriding AS*
- *AS-Path prepending*
- *EBGP policy requirement*
- *Timers*
- *Routing Reconfiguration*
  - *Route refresh*
- *BGP graceful restart capability*

## Aggregation

The main goal of aggregation is to summarize the number of network prefixes that are announced into the Internet. In fact, aggregation is a requirement when the mask length is too great. Your peers or the peers of your peers will filter some of them. They may want to reduce the number of prefixes.

However, the route aggregation can introduce some network loops or some black holes when it is not set properly.

---

### Note:

- A BGP router can advertise an aggregated network only if one route of the aggregate network is in the BGP table. For example if we consider four networks 192.168.0.0/24 through 192.168.3.0/24, the BGP router can advertise the aggregate network 192.168.0.0/22 only if at least one network (192.168.1.0/24 through 192.168.3.0/24) is in the BGP table.
  - If all the sub-networks of an aggregated network go down, this aggregated network will not be advertised.
  - It is recommended to check that the aggregated network is not stopped by an *Access List*.
-

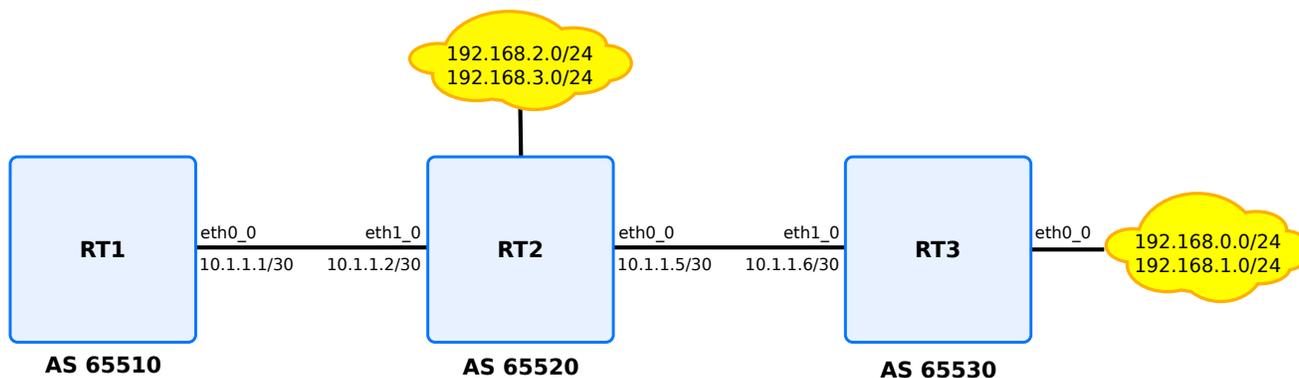


Fig. 4: BGP aggregation

The aggregation of the IPv4 network prefixes within the BGP tables can be done with the following command:

```
vrouter running bgp# address-family ipv4-unicast aggregate-address
                        PREFIX/M [summary-only true|false] [as-set true|false]
```

The aggregate command originates a new prefix. However, how to summarize the different AS-PATH ? There are two solutions:

- The AS-PATH is suppressed, although some network loops could be introduced.
- The AS-PATH is summarized within an unordered set (AS-SET), although some black hole could be created.

### No aggregation flags

When neither the `summary-only` flag nor the `as-set` flag are set, a route with the aggregated PREFIX/M is originated from the BGP router. However the sub-prefixes are still advertised.

### Example

```
routing bgp
  as 65500
  address-family
    ipv4-unicast
      network 192.168.3.0/24
      ..
      network 192.168.2.0/24
      ..
      aggregate-address 192.168.0.0/22
      ..
  ..
```

(continues on next page)

(continued from previous page)

```

neighbor 10.1.1.1
  remote-as 65510
..
neighbor 10.1.1.6
  remote-as 65530
..

```

After rt1 device peers with rt2, and rt2 peers with rt3, rt1 can receive following rib entries :

```

rt1> show bgp ipv4 unicast
BGP table version is 4, local router ID is 10.1.1.1, vrf id 0
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

```

| Network           | Next Hop | Metric | LocPrf | Weight | Path          |
|-------------------|----------|--------|--------|--------|---------------|
| *> 192.168.0.0/22 | 10.1.1.2 |        |        | 0      | 65520 i       |
| *> 192.168.0.0    | 10.1.1.2 |        |        | 0      | 65520 65530 i |
| *> 192.168.1.0    | 10.1.1.2 |        |        | 0      | 65520 65530 i |
| *> 192.168.2.0    | 10.1.1.2 | 0      |        | 0      | 65520 i       |
| *> 192.168.3.0    | 10.1.1.2 | 0      |        | 0      | 65520 i       |

Displayed 4 routes and 4 total paths

```

rt1> show bgp ipv4 unicast prefix 192.168.0.0/22
BGP routing table entry for 192.168.0.0/22
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  10.1.1.2
  65520, (aggregated by 65520 10.1.1.2)
  10.1.1.2 from 10.1.1.2 (10.1.1.2)
  Origin IGP, localpref 100, valid, external, atomic-aggregate, best
  AddPath ID: RX 0, TX 6
  Last update: Fri Sep 28 16:11:02 2018

```

---

#### Note:

- The aggregated prefix has the attribute atomic-aggregate, which means that the AS information is lost for the aggregate prefix (192.168.0.0/22).
  - Not to advertise the aggregated prefix, the flag summary-only can be set. Or a prefix-list or a distribute-list can be defined.
- 

Moreover this aggregated prefix is received by rt3 too.

```

rt3> show ipv4-route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

B>* 192.168.0.0/22 [20/0] via 10.1.1.5, ntfp2, 00:03:34
B>* 192.168.2.0/24 [20/0] via 10.1.1.5, ntfp2, 00:03:34
B>* 192.168.3.0/24 [20/0] via 10.1.1.5, ntfp2, 00:03:34

```

### Summary-only aggregation flag

When the summary-only flag is set and the as-set flag is not set, only the route with the aggregated PREFIX/M is originated from the BGP router. The sub-prefixes are not advertised. Moreover the ID of the router is set within the AS-PATH to help traffic engineering.

### Example

```

rt2 running bgp# address-family ipv4-unicast aggregate-address 192.168.0.0/22 summary-
<-only true

```

If the flag summary-only is set, the router will only advertise the aggregate prefix. We can notice that on the router which is advertising the aggregate prefix, the sub-prefixes have been suppressed, the remote peers will only see the aggregate prefix.

```

rt2> show bgp ipv4 unicast
BGP table version is 4, local router ID is 10.1.1.1, vrf id 0
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 192.168.0.0/22   0.0.0.0           32768 i
s> 192.168.0.0      10.1.1.6          0         0 65530 i
s> 192.168.1.0      10.1.1.6          0         0 65530 i
s> 192.168.2.0      0.0.0.0           0         32768 i
s> 192.168.3.0      0.0.0.0           0         32768 i

Displayed 5 routes and 5 total paths

```

The sub-prefixes which have been suppressed are labeled s.

On the remote peer, only the route to 192.168.0.0/22 is received by the BGP RIB (Routing Information Base).

```
rt1> show bgp ipv4 unicast
BGP table version is 4, local router ID is 10.1.1.1, vrf id 0
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

Network          Next Hop          Metric LocPrf Weight Path
*> 192.168.0.0/22  10.1.1.2                0 65520 i
```

However, rt3 is still getting the aggregated route.

```
rt1> show bgp ipv4 unicast
BGP table version is 4, local router ID is 10.1.1.1, vrf id 0
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

Network          Next Hop          Metric LocPrf Weight Path
*> 192.168.0.0/22  10.1.1.5                0 65520 i
*> 192.168.0.0     0.0.0.0            0      32768 i
*> 192.168.1.0     0.0.0.0            0      32768 i

Displayed 3 routes and 3 total paths
```

## As-set aggregation flag

When the summary-only flag is not set and the as-set flag is set, a route with the aggregated PREFIX/M is originated from the BGP router. Moreover the information of the previous AS-PATHs is collected into an unordered list called an AS-SET. This AS-SET, that is included within the new AS-PATH originated by the router, can help to avoid some networks loops. However the sub-prefixes are still advertised.

```
vrouter running bgp# address-family ipv4-unicast aggregate-address 192.168.0.0/22 as-
->set true
```

The AS information appears between brackets { }. It is an unordered list of the ASes.

In our example, if configured with as-set, rt2 can advertise an aggregate prefix because it knows at least one of its sub-networks.

Now by checking the rt2 BGP RIB we will see the as-set displayed. between brackets.

```

rt2> show bgp ipv4 unicast
BGP table version is 4, local router ID is 10.1.1.1, vrf id 0
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
                i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

Network          Next Hop          Metric LocPrf Weight Path
*> 192.168.0.0/22  0.0.0.0          32768 {65530} i
*> 192.168.0.0    10.1.1.6         0          0 65530 i
*> 192.168.1.0    10.1.1.6         0          0 65530 i
s> 192.168.2.0    0.0.0.0          0          32768 i
s> 192.168.3.0    0.0.0.0          0          32768 i

Displayed 5 routes and 5 total paths

```

### Combined summary-only and as-set aggregation flags

When both the `summary-only` and the `as-set` flags are set, a route with the aggregated PREFIX/M is originated from the BGP router. Moreover the information of the previous AS-PATHs is collected into an unordered list called an AS-SET. This AS-SET, that is included within the new AS-PATH originated by the router, can help to avoid some networks loops. The sub-prefixes are no longer advertised.

```

rt2 running bgp# address-family ipv4-unicast aggregate-address 192.168.0.0/22 summary-
↳only true
                    as-set true

```

By taking following example, rt1 will receive aggregated prefix with the as-set set.

```

rt2> show bgp ipv4 unicast
BGP table version is 4, local router ID is 10.1.1.1, vrf id 0
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
                i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

Network          Next Hop          Metric LocPrf Weight Path
*> 192.168.0.0/22  10.1.1.2         0          0 65520 {65530} i

```

### Confederation

A confederation is a set of many private ASes that are joined to be advertised as a single AS. A confederated AS is a confederation of many ASes that are joined by eBGP and that are themselves running an IGP.

The use cases are:

- a. Join independent ASes into a single AS.
- b. support multi-homed customers with a same ISP (Internet Service Provider).
- c. Avoid the scaling issues of the full-mesh eBGP routers.
- Configure a BGP confederation:

```
running bgp# confederation identifier 65501
```

- Join private ASes that belong to the same confederation:

```
running bgp# confederation peers 65502 peers 65501
```

### Example

Let's configure the following confederation:

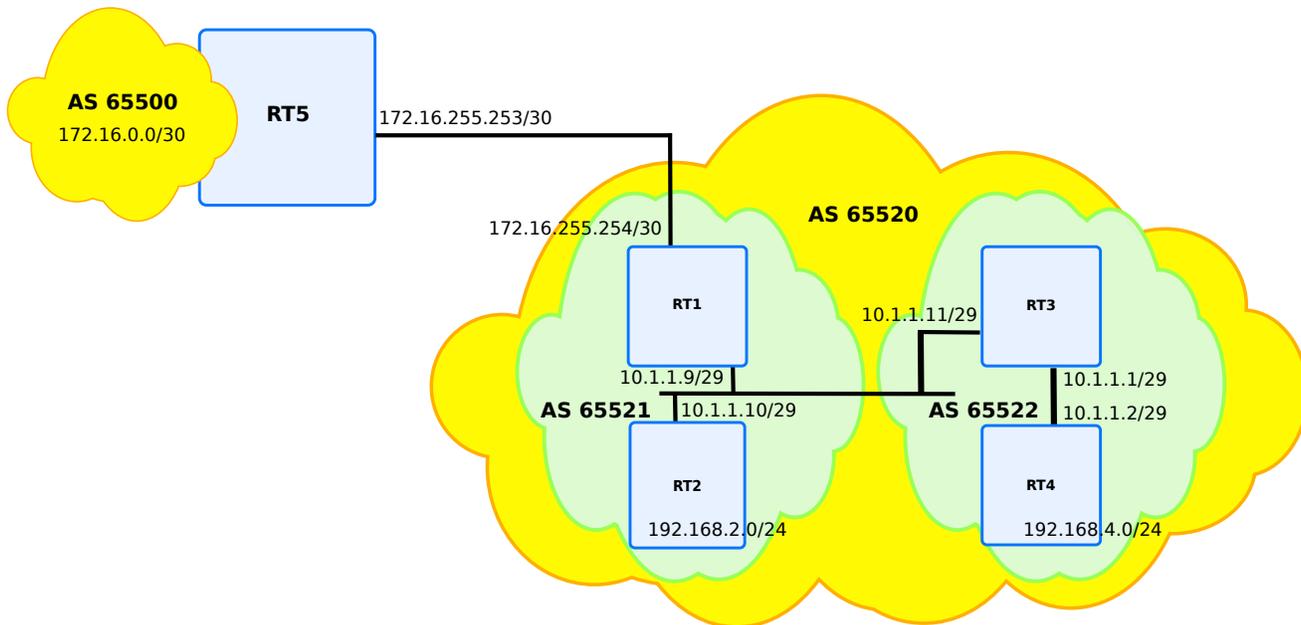


Fig. 5: BGP confederation

Where the following configurations are set:

**rt1**

```
vrf main
  interface physical eth0_0
    ipv4 address 10.1.1.9/29
    ..
  interface physical eth1_0
    ipv4 address 172.16.255.254/30
    ..
  routing bgp
    as 65521
    neighbor 10.1.1.11 remote-as 65522
    neighbor 10.1.1.11 address-family ipv4-unicast route-map out route-map-name_
↵change_nexthop
    neighbor 10.1.1.10 remote-as 65521
    neighbor 10.1.1.10 address-family ipv4-unicast route-map out route-map-name_
↵change_nexthop
    neighbor 172.16.255.253 remote-as 65500
    confederation identifier 65520
    confederation peers 65522
    ..
  ..
  ..
routing
  ipv4-access-list 1
    permit any
    ..
  ipv4-prefix-list filter
    seq 1 address 172.16.0.0/16 policy permit
    ..
  route-map change_nexthop
    seq 1 policy permit
    seq 1 match ip address prefix-list filter
    seq 1 set ip next-hop 10.1.1.9
    seq 2 policy permit
    seq 2 match ip address access-list 1
    ..
  ..
```

**rt2**

```
vrf main
  interface physical eth0_0
    ipv4 address 10.1.1.10/29
    ..
  interface physical eth1_0
    ipv4 address 192.168.2.1/24
    ..
  routing bgp
    as 65521
    neighbor 10.1.1.9 remote-as 65521
    confederation identifier 65520
    address-family ipv4-unicast network 192.168.2.0/24
    ..
  ..
```

**rt3**

```
vrf main
  interface physical eth0_0
    ipv4 address 10.1.1.11/29
    ..
  interface physical eth1_0
    ipv4 address 10.1.1.1/29
    ..
  interface loopback loop
    ipv4 address 192.168.3.1/24
    ..
  routing bgp
    as 65522
    neighbor 10.1.1.9 remote-as 65521
    neighbor 10.1.1.2 remote-as 65520
    confederation identifier 65520
    confederation peers 65521
    address-family ipv4-unicast network 192.168.3.0/24
    ..
  ..
```

**rt4**

```
vrf main
  interface physical eth0_0
    ipv4 address 192.168.4.1/24
    ..
  interface physical eth1_0
    ipv4 address 10.1.1.2/29
    ..
  routing bgp
    as 65522
    neighbor 10.1.1.1 remote-as 65522
    confederation identifier 65520
    address-family ipv4-unicast network 192.168.4.0/24
    ..
  ..
```

**rt5**

However, when rt5 peers with rt1, it peers to the AS 65520 that is rt1's BGP confederation identifier. It does not peer to the AS 65521 that is internal to the AS 65520:

```
vrf main
  interface physical eth0_0
    ipv4 address 172.16.0.1/16
    ..
  interface physical eth1_0
    ipv4 address 172.16.255.253/30
    ..
  routing bgp
    as 65000
    neighbor 172.16.255.254 remote-as 65522
    address-family ipv4-unicast network 172.16.0.0/16
    ..
  ..
```

- Check this configuration on rt3 that displays the confederation path between parenthesis. The fib can also be dumped.

```
rt3> show bgp ipv4 unicast
BGP table version is 2, local router ID is 192.168.3.1, vrf id 0
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Next-hop codes: @NNN next-hop's vrf id, < announce-nh-self
```

(continues on next page)

(continued from previous page)

Origin codes: i - IGP, e - EGP, ? - incomplete

| Network        | Next Hop  | Metric | LocPrf | Weight | Path            |
|----------------|-----------|--------|--------|--------|-----------------|
| 172.16.0.0     | 10.1.1.9  | 0      | 100    | 0      | (65521) 65500 i |
| *> 192.168.2.0 | 10.1.1.10 | 0      | 100    | 0      | (65521) i       |
| *> 192.168.3.0 | 0.0.0.0   | 0      |        | 32768  | i               |
| *>i192.168.4.0 | 10.1.1.2  | 0      | 100    | 0      | i               |

Displayed 3 routes and 3 total paths

**rt3>** show bgp ipv4 unicast prefix 172.16.0.0/16

BGP routing table entry for 172.16.0.0/16

Paths: (1 available, no best path)

Advertised to non peer-group peers:

10.1.1.9

(65521) 65500

10.1.1.9 from 10.1.1.9 (172.16.255.254)

Origin IGP, metric 0, localpref 100, invalid, confed-external, best

AddPath ID: RX 0, TX 22

Last update: Fri Oct 12 09:34:14 2018

The FIB (Forwarding Information Base) can also be dumped:

**rt3>** show ipv4-routes

Codes: K - kernel route, C - connected, S - static, R - RIP,

O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,

T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,

F - PBR, f - OpenFabric,

> - selected route, \* - FIB route, q - queued route, r - rejected route

C>\* 10.1.1.0/29 is directly connected, eth0\_0, 00:23:26

C>\* 10.1.1.8/29 is directly connected, eth0\_0, 00:23:26

B>\* 172.16.0.0/16 [200/0] via 10.1.1.9, eth0\_0, 00:18:11

B>\* 192.168.2.0/24 [200/0] via 10.1.1.10, eth0\_0, 00:17:17

C>\* 192.168.3.0/24 is directly connected, loopback, 00:23:26

B>\* 192.168.4.0/24 [200/0] via 10.1.1.2, eth1\_0, 00:17:17

**Note:** if a route-map had not been added to rt1, 172.16.0.0/16 would not have been visible in rt3, because it has no route to 172.16.255.253. It is a feature of BGP that requires to work with an IGP to resolve the recursive routes that do not have a directly connected gateway. Moreover, it means that the eBGP sessions between the confederation sub-ASes do not change the next hop attribute.

For example, you could add RIP or OSPF v2 on rt1, rt2, rt3 and rt4 that will be the IGP of all the AS65520.

---

## Overriding AS

When working with both public BGP peers and private BGP peers, it is wished to have one single BGP instance, and in the same time, having the ability to override the default AS value. This can be done by using local-as value, where it is possible to override default AS value by the one that is set as local-as value.

Following configuration illustrates what the configuration could be. real AS value (65000 here) is hidden behind 64512. Remote peer only sees 64512 value.

```
vrf main
  routing bgp
    as 65000
    neighbor 10.125.0.2 remote-as 64622
    neighbor 10.125.0.2 local-as as-number 64512 no-prepend true replace-as true
    ..
  ..
  ..
```

## AS-Path prepending

On some situations, it is also wished to modify the as-path list. For instance, on transit routers, the as-path list may be enlarged in order to influence incoming traffic. Actually, by increasing the as-path list size, BGP best path selection algorithm may pick up the routers with the shortest as-path list.

The following route-map configuration can be applied to outgoing prefixes exchanged with BGP peers. as-path prepending action will prepend as-path values to the original as-path list. The priority number configured will determine which as-path value to insert first.

For instance, below route-map will prepend {65500, 65100} in the as-path list following the configured order 10, 20.

```
vrf main
  routing bgp as 65500
  ..
routing
  ipv4-prefix-list blocka
    seq 10 address 10.0.0.0/8 policy permit
    ..
  route-map bgp-export-block
    seq 10
      policy permit
```

(continues on next page)

(continued from previous page)

```
match
  ip
    address
      prefix-list blocka
      ..
    ..
  set
    as-path
      prepend
        asn 20
          65100
          ..
        asn 10
          65500
          ..
      ..
    ..
  ip
    next-hop 184.106.55.69
```

## EBGP policy requirement

When interoperating with eBGP peers, route propagation may become riskier if no policies are set up on those peers. **RFC 8212** (<https://tools.ietf.org/html/rfc8212.html>) enforces that policy by checking that incoming and outgoing filters are applied for eBGP sessions. With this policy, no route will be either accepted (if no incoming filter) nor announced (if no outgoing filter). Below command can be used to enforce the behavior:

```
vrouter running bgp# ebgp-requires-policy true
```

## Timers

The BGP timers are specific to the neighbors.

- Set specific timers:

```
vrouter running bgp# neighbor 10.125.0.3 timers keepalive-interval 15 hold-time 30
```

---

**Tip:** A good practice is to configure the same value on both sides of the TCP connection. Generally, these values should not be changed; however when the processing time of the BGP table is too long for the CPU to fire the keepalive timer, the later could be increased.

---

## Routing Reconfiguration

Some configuration items may need the BGP routing tables to be refreshed. This is the case for multipath configuration. Enabling multipath needs to analyse all the routing table to see if there are ECMP entries.

BGP provides 2 mechanisms to permit this refresh:

- either by issuing BGP route refresh messages to remote peers. This message asks remote peer to send back all BGP updates for a defined (AFI (Address Family Identifier), SAFI) address-family.
- or by enhancing software reconfiguration inbound. An inbound RIB is created for each peer, for a defined (AFI, SAFI). This is the ADJ-RIB-IN. All incoming BGP updates are stored in ADJ-RIB-IN and are kept unmodified. This permits reinjecting original BGP updates of remote peer, when needed. Enhancing software reconfiguration inbound can be configured on each address-family node.

The routing reconfiguration will be automatically triggered upon some reconfiguration elements. If software reconfiguration is not configured, then default behaviour will issue a route refresh message with remote peer.

Anytime, ADJ-RIB-IN can be flushed by using a `flush` command. This will force to rebuild the ADJ-RIB-IN command by issuing update with remote peer:

```
flush bgp vrf main all soft in
```

## Route refresh

Route refresh is an extension to BGP that is defined in **RFC 2918** (<https://tools.ietf.org/html/rfc2918.html>). Using this feature, a BGP router can request a complete retransmission of the peer's routing information without tearing down and reestablishing the BGP session, saving a route flap. It is used to facilitate routing policy changes, without storing an unmodified copy of the peer's routes on the local router to save memory. The capability must be supported by both routers of a BGP session. When both routers in the peering session support this extension, each router will respond to requests issued from the peer without operator intervention.

Route Refresh is enabled by default.

When the command `flush` is used, Route Refresh messages are sent to the peers, the router receives one or more Update packets with all the routes of the Adj-RIB-Out.

## Example

Let's configure the following peering:

```
routing bgp
  as 65000
  neighbor 172.16.255.254 remote-as 65522
  address-family ipv4-unicast network 172.16.0.0/16
  .. .. ..
```

Then the peering happens. And the RIB is feeded with remote updates from remote. No need to configure the multipath feature, since it is enabled by default.

The local peer will mark as staled the local entries learnt from the remote peer, then will send a BGP refresh message to the remote peer. The remote peer will send back the BGP updates, and the local instance will refresh the RIB accordingly.

### BGP graceful restart capability

Usually when BGP on a router restarts, all the BGP peers detect that the session went down, and then came up. This “down/up” transition results in a “routing flap” and causes BGP route re-computation, generation of BGP routing updates and flap the forwarding tables. It could spread across multiple routing domains. Such routing flaps may create transient forwarding blackholes and/or transient forwarding loops. They also consume resources on the control plane of the routers affected by the flap. As such they are detrimental to the overall network performance.

This feature proposes a mechanism for BGP that would help minimize the negative effects on routing caused by BGP restart. The graceful restart capabilities (code-64) will be exchanged between the BGP speakers through the open messages. Routes advertised by the restarting speaker will become stale in the peer speakers’ routing table. On expiry of `restart time` the stale routes will be deleted if the restarting speaker does not come up. Once the restarting speaker re-establish the BGP session within the `restart time` the stale routes will be converted to normal routes. Traffic flow through the stale routes will not be stopped while the BGP speaker is restarting.

- Enable BGP graceful restart:

```
vrouter running bgp# graceful-restart restart-time 60
```

```
vrouter running bgp# graceful-restart stalepath-time 120
```

### BGP security

BGP is used for inter-domain routing, so it is a critical service for the Internet infrastructure. Therefore security aspect of BGP, with valid routing advertisement, is a high issue and the current system is highly vulnerable to human errors, as well as a wide range of attacks.

Filtering is currently the most used mechanism. Nevertheless complementary security features may be used to add security with BGP. Thus, in some cases MD5 authentication may be used to control BGP routing information advertisement, as described for OSPF.

- *BGP filtering*
  - *Configuring a BGP-4 distribute list*
  - *Configuring a BGP-4 prefix list*
  - *Communities Filters*

- *BGP Authentication*

## BGP filtering

Two types of BGP filtering method exist:

**Distribute-list** Allows filtering on prefix basis,

**AS-PATH access-list** Filters all networks in relation with a particular ASN.

## Configuring a BGP-4 distribute list

Once an IPv4 *Access List* is created, it is possible to apply this access-list to a neighbor. The number of prefixes will be modified/filtered so that the neighbor will not see the exact entries that local device sees.

Following configuration illustrates 2 devices rt1 and rt2, where rt2 is configured to apply filtering by using distribute list.

### rt1

```
routing bgp
  router-id 10.1.1.1
  as 65510
  neighbor 10.1.1.2 remote-as 65520
```

### rt2

```
routing
  ipv4-access-list acl_name
    remark description
    deny 192.168.1.0/24
    permit any
    ..
  ..
vrf main
routing bgp
  router-id 10.1.1.2
  address-family ipv4-unicast network 192.168.1.0/24
  .. .. ..
  address-family ipv4-unicast network 192.168.2.0/24
  .. .. ..
```

(continues on next page)

(continued from previous page)

```

as 65520
neighbor 10.1.1.1 remote-as 65510
neighbor 10.1.1.1 address-family ipv4-unicast distribute-list out access-list acl_
↪name
..
..
..

```

Consequently, 192.168.1.0/24 prefix will not be exported to rt1

```

rt1> show bgp ipv4 unicast
BGP table version is 40, local router ID is 10.1.1.1, vrf id 0
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop          Metric LocPrf Weight Path
*> 192.168.2.0    10.1.1.2          0           0 65520 i

```

---

**Note:** The below IPv4 prefix-list should be preferred to the IPv4 access-lists.

---

### Configuring a BGP-4 prefix list

1. Define the prefix-list rule as per *Prefix List*.
2. Apply the prefix list rule to a neighbor:

```
neighbor 10.125.0.3 address-family ipv4-unicast prefix-list in prefix-list-name pname
```

### Communities Filters

The attribute `community` permits to group destinations in a community and apply routing decisions. It is an optional, global transitive attribute in the numerical range of 1 to 4,294,967,200. Based on the community, you can control the routing information. In BGP there are some predefined well known communities which are:

**no-export** The routes of this community must not be advertised to external peer. Value is 0xFFFFFFFF01.

**no-advertise** The routes must not be advertised to any peer. Value is 0xFFFFFFFF02.

**internet** The routes may be advertised to any peer. Value is 0x0.

**local-as** Used in confederation to avoid sending packets outside the local AS. Value is 0xFFFFFFFF03.

In general, BGP community has the form of AS:NN where AS is the autonomous system number, and NN is a number.

In addition to communities, BGP introduced one new kind of communities : extended communities. It extends the range of values. For instance, extended community can be used to store 4 AS byte value, while community is generally used to encode only 2 AS value. The format changed compared with community, because the services are different ( for instance, L3VPN services benefit from route target extended communities).

Both communities are used to apply filtering on incoming or outgoing BGP updates. This can be done by using route-maps.

---

**Note:** To match a community or an extended community attribute it is recommended to use route-maps. In general, BGP community has the form of AS:NN where AS is the autonomous system number, and NN is a number. Conversely, BGP extended community is 6 octet wide, and has three available forms : as2B:NNNN or as4B:NN or as2B:IPv4. Where as2B and as4B respectively stand for the autonomous system 2 byte and 4 byte AS value . NN and NNNN are arbitrary value encoded with 2 and 4 byte values.

---

The community and extended attribute is sent to neighbors by default with the option both (standard and extended community):

```
neighbor 10.125.0.3 address-family ipv4-unicast send-community both
```

- Delete the community parameters:

```
neighbor A.B.C.D address-family ipv4-unicast
del send-community
```

The community's or extended community's policies are examined in the priority order for each prefix. As soon as a policy matches a prefix, the desired behaviour (permit or deny) is applied (when used in a "match community id <community name>" clause of a route-map: it's a match/it's not a match) and the processing stops for this prefix. There is also an implicit final deny policy in each community list that rejects any prefix that did not match any previous defined policies. A policy applies to a prefix if and only if all of its communities are set on the prefix.

More information about how to configure BGP communities lists and BGP extended communities lists can be found below.

## Community list

Community list is a group of rules which permit to filter or set attributes based on different lists of community numbers.

Community list has two types. It could be either a standard community list or an expanded community list.

Standard community list defines communities attributes. Thus, it will be directly compared to BGP communities attribute in BGP updates.

On the other hand, expanded community list defines its communities attribute in a string with regular expression.

A community list is used in a match clause of a *route map*. To illustrate a use case, prefixes learnt from various transit providers may bring such information per prefix. It may be desirable to append its own community tag based on the incoming community tags already present. The syntax of community list usage in route-map is the following one:

```
routing
route-map rmap_name
seq 11 policy permit|deny
seq 11 match community id com_name exact-match true
```

For example, the following configuration will redistribute any prefix having at least one of the communities 22850:65101 or 22850:65102:

```
routing
  route-map myrmap
    seq 1
      policy permit
      match
        community id MYCLIST
      ..
  bgp
    community-list MYCLIST
      policy 1 permit 22850:65101
      policy 2 permit 22850:65102
```

But, the following configuration will redistribute any prefix having both communities 22850:65101 and 22850:65102:

```
routing
  route-map myrmap
    seq 1
      policy permit
      match
        community id MYCLIST
      ..
  bgp
    community-list MYCLIST
      policy 1 permit 22850:65101 22850:65102
```

## Extended Community list

An extended community list is a group of rules which permit to filter or set attributes based on different lists of extended community numbers.

Extended community list has two types: standard and expanded extcommunity lists.

A standard extended community list is based on BGP extended community attribute. Two kinds of communities can be created : route-target ( RT (Route Target)), and site-of-origin ( SOO (Site Of Origin)). The former is used to define import and export policies across the vrfs, while the latter is used to prevent routing loops between sites.

An expanded extended community list uses regular expression to match extended communities attribute in BGP updates.

An extended community list is used in a match clause of a *route map*. Like for community lists, extended community lists can be used for receiving prefixes from transit provider, that need to be appended with some extended communities tags accordingly. The syntax of extended community list usage in route-map is the following one:

```
routing
route-map rmap_name
seq 11 policy permit|deny
seq 11 match extcommunity ecom_name_1

extcommunity-list ecom_name_1 policy permit soo 65501:43
extcommunity-list ecom_name_2 policy deny rt 10.125.0.1:54
```

For example, the following configuration will redistribute any prefix having at least one of the extended communities soo 65501:43 or rt 10.125.0.1:54:

```
routing
  route-map myrmap
    seq 1
      policy permit
      match
        extcommunity MYXCLIST
        ..
  bgp
    extcommunity-list MYXCLIST
    policy 1 permit soo 65501:43
    policy 2 permit rt 10.125.0.1:54
```

But, the following configuration will redistribute any prefix having both extended communities soo 65501:43 and rt 10.125.0.1:54:

```
routing
  route-map myrmap
    seq 1
```

(continues on next page)

(continued from previous page)

```
    policy permit
    match
        extcommunity MYXCLIST
        ..
bgp
    extcommunity-list MYXCLIST
    policy 1 permit soo 65501:43 rt 10.125.0.1:54
```

## BGP Authentication

BGP authentication is using MD5 (Message Digest 5). This feature relies on the Operating System support for the TCP MD5 signature option as proposed in the [RFC 2385](https://tools.ietf.org/html/rfc2385.html) (https://tools.ietf.org/html/rfc2385.html). This OS option is used with the BSD-like configuration API.

The command format for BGP MD5 is as follows:

```
vrf main
    routing bgp
        neighbor 10.125.0.1 password my-secret
```

For information, when analyzing the BGP packets with the sniffer `traffic-capture`, it is possible to verify that the option is taken into account.

## BGP Flowspec

### Overview

BGP flowspec introduces a new Network Layer Reachability Information (NLRI) encoding format that is used to distribute traffic rule flow specifications. Basically, instead of simply relying on destination IP address for IP prefixes, the IP prefix is replaced by a n-tuple consisting of a rule. That rule can be a more or less complex combination of the following:

All below items are supported in this release.

- Network IP source/destination (can be one or the other, or both), for both IPv4 and IPv6.
- Layer 4 information for UDP (User Datagram Protocol), TCP : source port, or destination port, or any port for both IPv4 and IPv6.
- Layer 4 information for ICMP type and ICMP code, for both IPv4 and IPv6.
- Layer 3 information : DSCP (Differentiated Services Code Point) value, Protocol type, packet length for both IPv4 and IPv6.
- Layer 3 information : fragmentation for IPv4 support

- Misc layer 4 TCP flags, for both IPv4 and IPv6.

A combination of the above rules is applied for traffic filtering. This is encoded as part of specific BGP extended communities and the action can range from the obvious rerouting (to nexthop or to separate VRF) to shaping, or discard.

Following IETF (Internet Engineering Task Force) RFC (Request For Comment) documents have been used to implement flowspec:

- **RFC 5575** (<https://tools.ietf.org/html/rfc5575.html>)
- **Draft Flowspec Redirect IP** (<https://tools.ietf.org/id/draft-ietf-idr-flowspec-redirect-ip-02.txt>)
- **Draft Flowspec IPv6** (<https://tools.ietf.org/html/draft-ietf-idr-flow-spec-v6-09>)

## Configuration guide

In order to configure an IPv4 flowspec engine, use the following configuration. As of today, it is only possible to configure flowspec on default VRF. To enter the BGP flowspec sub-context:

```
vrouter running bgp# neighbor A.B.C.D remote-as AS
vrouter running bgp# neighbor A.B.C.D address-family ipv4-flowspec
vrouter running ipv4-flowspec# enabled true
```

**AS** The remote Autonomous system ID associated with neighbor

**A.B.C.D** The remote BGP peer to peer with BGP flowspec address family support

Exemple:

```
routing bgp
  as 5
  neighbor 1.0.0.1 remote-as 2
  neighbor 1.0.0.1 address-family ipv4 flowspec
  ..
```

In a similar way, following extract illustrates how IPv6 flowspec engine can be set:

```
vrf main
  routing bgp
    as 65500
    neighbor 10.125.0.2 remote-as AS
    neighbor 10.125.0.2 address-family ipv6-flowspec enabled true
```

## Flowspec Per Interface

One nice feature to use is the ability to apply flowspec to a specific interface, instead of applying it to the whole machine. Despite the following IETF draft `idr flowspec interface set` (<https://tools.ietf.org/html/draft-ietf-idr-flowspec-interfaceset-03>) is not implemented, it is possible to manually limit flowspec application to some incoming interfaces. Actually, not using it can result to some unexpected behaviour like accounting twice the traffic, or slow down the traffic (filtering costs). To limit flowspec to one specific interface, use the following command, under BGP flowspec family.

```
routing bgp
  address-family ipv4-flowspec
    enabled true
    local-install eth1
```

By default, Flowspec is activated on all interfaces. Installing it to a named interface will result in allowing only this interface. Reversely, enabling any interface will flush all previously configured interfaces.

## Flowspec redirect IP

Flowspec provides also the ability for traffic to be redirected according to nexthop IP information. BGP flowspec entries have a BGP extended community option, that tells that the flowspec information should be redirected to the IP contained in the nexthop attribute of the BGP update received. Using that option to redirect traffic simply consists in ensuring that the IP information is reachable through using the routing table logic. For instance, create a static route

```
vrf main
  routing static ipv4-route 2.2.2.2/32 next-hop 10.1.2.3
```

## Flowspec redirect VRF

An other nice feature to configure is the ability to redirect traffic to a separate VRF. This feature does not go against the ability to configure Flowspec only on default VRF. Actually, when you receive incoming BGP flowspec entries on that default VRF, you can redirect traffic to an other VRF.

As remind, BGP flowspec entries have a BGP extended community that contains an RT, that is to say a route target. Finding out a local VRF based on route target consists in the following:

- A configuration of each VRF must be done, with its RT set

Each VRF is being configured within a BGP VRF instance with its own RT list. RT is defined in **RFC 4364** (<https://tools.ietf.org/html/rfc4364.html>) and is an attribute associated to a VRF. In the VRF context, incoming route entries have their own RT, and incoming BGP instance selects for which VRF the incoming entry is imported, thanks to RT. route entries can be duplicated, if one route target matches several VRs. In the flowspec context, only the first VRF matching the incoming flowpsec entry will be selected. The RT is encoded as BGP extended communities and is 8 byte long. The first 2 byte contain the format of the RT, while the last 6 byte define the

values of the RT. Accepted format matches the following: A.B.C.D:U16, or U16:U32, U32:U16. U32 and U16 respectively stand for 4 byte integer value and 2 byte short value. Values can either be mapped to ASnumber of VXLAN identifier in case of overlay with vxlan tunnels. A.B.C.D stands for an IP address, and can be mapped to bgp router identifier or tunnel endpoint.

As said before, a VRF can have a list of route targets. To configure the RT list, use the following command under BGP ipv4 unicast family:

```
vrf main
  routing bgp
    router-id 1.0.0.2
    as 65500
    neighbor 1.0.0.1 remote-as 65100
    neighbor 1.0.0.1 address-family ipv4-flowspec enabled true
    ..
    ..
  ..
vrf monitor
  routing bgp
    as 65500
    address-family ipv4-unicast
    route-target redirect-import 11:22
```

In order to illustrate, if the route target configured in the flowspec entry is `10.1.1.2:65200`, then a BGP instance of a specific VRF with the same route target will be set. That VRF will then be selected. The below full configuration example depicts how route targets are configured and how VRF and interfaces configuration is done. Note that the VRs are mapped on Linux Network Namespaces. For convey traffic through VRs, Cross-VRF interfaces are needed. Basically, those are veth pair interfaces with specific properties, and without IP attributes. More information in *XVRF Interface types*.

```
router> show config
vrf main
  interface xvrf monitor
    enabled true
    link-interface main
    link-vrf monitor
    ..
  ..
  routing
    bgp
      router-id 192.168.0.162
      as 65100
      neighbor 192.168.0.161
        remote-as 65100
      address-family
        ipv4-flowspec
```

(continues on next page)

(continued from previous page)

```

    ..
    ..
    ..
    ..
vrf monitor
  interface xvrf main
    link-interface monitor
    link-vrf main
    ..
    ..
  routing
    bgp
      as 65200
      address-family
        ipv4-unicast
          route-target
            redirect-import 11:22
            redirect-import 10.1.1.2:65200
            redirect-import 10.1.1.2:65100
            ..
          ..
        ..
      ..
    ..
  ..

```

### Flowspec redirect VRF for IPv6

In a similar way, BGP flowspec for IPv6 uses specific ipv6 extended communities attributes, as defined in **RFC 5701** (<https://tools.ietf.org/html/rfc5701.html>). The format of the community attribute is 20 bytes length, instead of 8 bytes length, and as such, can be used to store an IPv6 address and 2 additional extra-bytes. The format is defined as follows :<IPv6>:AS2B.

As defined in [Dissemination of Flow Specification Rules for IPv6](https://tools.ietf.org/html/draft-ietf-idr-flow-spec-v6-09) (<https://tools.ietf.org/html/draft-ietf-idr-flow-spec-v6-09>), the BGP NLRI is sent with an extended IPv6 community attribute with community type set to 0x00 and community sub type set to 0x0c. The BGP daemon reads the incoming parameters and compares the extended IPv6 community with locally configured IPv6 route targets. The first IPv6 RT matching the incoming extended community will represent the bgp VRF instance where traffic will be redirected to. Below example illustrates how this can be configured:

```

vrf main
  routing bgp
    router-id 1.0.0.2
    as 65500

```

(continues on next page)

(continued from previous page)

```

neighbor 1.0.0.1 remote-as 65100
neighbor 1.0.0.1 address-family ipv6-flowspec enabled true
..
..
..
vrf monitor
routing bgp
as 65500
address-family ipv6-unicast
ipv6-route-target redirect-import 11::22::0:0

```

### Flowspec Monitor and troubleshooting

You can monitor policy-routing objects by using one of the following commands. Those command rely on the filtering contexts configured from BGP, and get the statistics information retrieved from the underlying system. In other words, those statistics are retrieved from linux netfilter.

```
rt1> show bgp pbr ipset
```

About rule contexts, it is possible to know which rule has been configured to policy-route some specific traffic. The first table identifier displayed on the former `show bgp pbr iptable` command can be used in the latter command to know about routing information.

```
rt1> show bgp pbr iptable
```

You can troubleshoot BGP flowspec, or BGP policy based routing. Ensuring that a flowspec entry has been correctly installed and that incoming traffic is policy-routed correctly can be checked like illustrated below. First of all, you must check whether the flowspec entry has been installed or not.

```

rt1> show bgp ipv4 flowspec prefix 5.5.5.2/32
BGP flowspec entry: (flags 0x418)
Destination Address 5.5.5.2/32
IP Protocol = 17
Destination Port >= 50 , <= 90
FS:redirect VRF RT:255.255.255.255:255
received for 18:41:37
installed in PBR (match0x271ce00)

```

This means that the flowspec entry has been installed in a linux iptable named `match0x271ce00`. Once you have confirmation it is installed, you can check whether you find the associate entry by executing following command. You can also check whether incoming traffic has been matched by looking at counter line.

```

rt1> show bgp pbr ipset set match0x271ce00
  IPset match0x271ce00 type net,port
    to 5.5.5.0/24:proto 6:80-120 (8)
      pkts 1000, bytes 1000000
    to 5.5.5.2:proto 17:50-90 (5)
      pkts 1692918, bytes 157441374

```

As you can see, the entry is present. Note that a linux iptable can be used to host several BGP flowspec entries.

In order to know where the matching traffic is redirected to, you have to look at the policy routing rules. The policy-routing is done by forwarding traffic to a routing table number. That routing table number is reached by using a linux iptable. The relationship between the routing table number and the incoming traffic is a MARKER that is set by the iptable referencing the linux ipset. In flowspec case, linux iptable referencing the linux ipset context have the same name.

So it is easy to know which routing table is used by issuing following command:

```

rt1> show bgp pbr iptable
  IPtable match0x271ce00 action redirect (5)
    pkts 17000000, bytes 158000000
    table 257, fwmark 257
  ...

```

This allows to see where the traffic is forwarded to. Actually, in the case of redirect VRF action, a route leak has been pushed to reach a separate VRF. Example below depicts what a route to VRF monitor looks like, since a default route in table 257 has been installed to reach monitor.

```

router> show ipv4-routes table 257
  Codes: K - kernel route, C - connected, S - static, R - RIP,
  O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
  T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
  F - PBR, f - OpenFabric,
  > - selected route, * - FIB route, q - queued route, r - rejected route
B>* 0.0.0.0/0 [20/0] is directly connected, monitor, 19:07:48

router> show ipv4-routes vrf monitor
  Codes: K - kernel route, C - connected, S - static, R - RIP,
  O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
  T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
  F - PBR, f - OpenFabric,
  > - selected route, * - FIB route, q - queued route, r - rejected route
VRF monitor:
S>* 0.0.0.0/0 [1/0] via 1.1.1.2, eth1, 19:25:04
C>* 1.1.1.0/24 is directly connected, eth1, 19:25:05

```

## BGP in virtual routers

### BGP configuration and monitoring in VRF

Usually, BGP router is configured in VR main. To handle virtual routers, a separate VRF can be specified. The routes learnt and configured will be stored in the corresponding forwarding tables. It is possible to create multiple BGP instances on the same machine.

- Create a BGP instance on a VRF named customer1, by using following command:

```
vrf customer1
  routing bgp
    as 54
  ..
  ..
```

Then you can continue the configuration as usual. The BGP peering and the redistribution will happen on the whole VRF. All configured interfaces with addresses, and routing information on that VRF will be used.

To get routing information about BGP in that VR instance, you can use following command, that will dump all the instances configured.

```
vrouter> show bgp vrfs
Type Id      routerId      #PeersVfg  #PeersEstb  Name  L3-VNI  Rmac
DFLT 0       192.168.0.162  2          1  Default  0       00:00:00:00:00:00
VRF  2       0.0.0.0       0          0  customer1  0       00:00:00:00:00:00
VRF  3       1.1.1.1       0          0  customer2  0       00:00:00:00:00:00
```

To get more information on a specific VRF, you can use following command, and the VRF name to get routing information:

```
vrouter> show bgp vrf customer1 ipv4 unicast
BGP table version is 2, local router ID is 1.1.1.1, vrf id 2
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete
Network      Next Hop      Metric LocPrf Weight Path
*> 3.3.3.0/24  0.0.0.0       0      32768 i
*> 4.4.4.0/24  0.0.0.0       0      32768 i

Displayed 2 routes and 2 total paths
```

## BGP use case for VRF

A common use case is to provide a per customer BGP peering. This use case can happen, when several customers share physical resources of a machine, but are isolated by means of either physical interfaces or VLAN interfaces. The following configuration gives an overview on how to create multiple BGP instances tighted with VLAN interfaces.

As you can see on below example, 2 instances of BGP are created, each one run over VLAN interface with its own peer. The same autonomous system can be used for all the instances. The BGP contexts share the same system process but will not share the same forwarding information.

```
vrf customer1
  routing bgp
    as 65555
    router-id 192.168.1.1
    neighbor 192.168.1.2 remote-as 65555
    ..
    ..
  interface vlan vlan10
    vlan-id 10
    link-interface eth0_0
    ipv4 address 192.168.1.1/24
    ..
    ..
  ..
vrf customer2
  routing bgp
    as 65555
    router-id 192.168.2.1
    neighbor 192.168.2.2 remote-as 65555
    ..
    ..
  interface vlan vlan20
    vlan-id 20
    link-interface eth0_0
    ipv4 address 192.168.2.0/24
    ..
    ..
```

---

**Note:** With the above example, it could have been possible to use the same IP mapping for both routing entities. An other benefit of having separate entities is that IP mapping can overlap.

---

## BGP use case for cross-VRF routes

An other common use case is to install cross-VRF routes between 2 BGP instances. A proposal consists in using a mesh of BGP peerings via veth links. Below example illustrates what can be done to install cross VRF routes between admin vrf and service vrf.

```
vrf admin
  interface physical eth0
    port pci-b0s5
    ipv4 address 192.168.2.1/24
    .. ..
  interface veth service1
    link-interface admin
    link-vrf service1
    .. ..
  interface veth service2
    link-interface admin
    link-vrf service2
    .. ..
  interface loopback localip
    ipv4 address 169.254.0.100/32
    .. ..
  routing static ipv4-route 169.254.0.101/32 next-hop service1
  routing static ipv4-route 169.254.0.102/32 next-hop service2
  routing bgp
    as 11
    ebgp-connected-route-check false
    router-id 192.168.2.1
    address-family ipv4-unicast
      redistribute connected route-map nocalocal
      .. ..
    neighbor 192.168.2.2
      remote-as 12
      address-family ipv4-unicast as-outbound-update
        action remove as-type all
      .. ..
    neighbor-group local update-source localip
  neighbor 169.254.0.101
    remote-as 64601
    local-as as-number 64600 no-prepend true replace-as true
    neighbor-group local
    ..
  neighbor 169.254.0.102
    remote-as 64602
    local-as as-number 64600 no-prepend true replace-as true
```

(continues on next page)

(continued from previous page)

```
    neighbor-group local
    ..
    ..
    ..
vrf service1
  interface physical eth1
    port pci-b0s6
    ipv4 address 10.1.2.1/24
    .. ..
  interface veth admin
    link-interface service1
    link-vrf admin
    .. ..
  interface loopback localip
    ipv4 address 169.254.0.101/32
    .. ..
  routing static ipv4-route 169.254.0.100/32 next-hop admin
  routing bgp
    as 11
    ebgp-connected-route-check false
    router-id 10.1.2.1
    address-family ipv4-unicast
      redistribute connected route-map nlocal
    .. ..
  neighbor 10.1.2.2
    remote-as 13
    address-family ipv4-unicast as-outbound-update
      action remove as-type all
    .. ..
  neighbor-group local update-source localip
  neighbor 169.254.0.100
    remote-as 64600
    local-as as-number 64601 no-prepend true replace-as true
    neighbor-group local
    ..
    ..
    ..
vrf service2
  interface physical eth2
    port pci-b0s4
    ipv4 address 10.2.2.1/24
    .. ..
```

(continues on next page)

(continued from previous page)

```

interface veth admin
  link-interface service2
  link-vrf admin
  .. ..
interface loopback localip
  ipv4 address 169.254.0.102/32
  .. ..
routing static ipv4-route 169.254.0.100/32 next-hop admin
routing bgp
  as 11
  ebgp-connected-route-check false
  router-id 10.2.2.1
  address-family ipv4-unicast
    redistribute connected route-map nlocal
    .. ..
  neighbor 10.2.2.2
    remote-as 14
    address-family ipv4-unicast as-outbound-update
      action remove as-type all
    .. ..
  neighbor-group local update-source localip
  neighbor 169.254.0.100
    remote-as 64600
    local-as as-number 64602 no-prepend true replace-as true
    neighbor-group local
    ..
  ..
  ..
routing ipv4-access-list local seq 1 permit 169.254.0.0/24
routing route-map nlocal
  seq 1
    policy deny
    match ip address access-list local
    ..
  seq 2
    policy permit
    ..
  ..
  ..

```

In addition to establish 2 eBGP sessions with external devices, an internal BGP peering is established between 3 VRs. Below dump gives an overview of the BGP sessions from VRF admin perspective.

```
rt1> show bgp vrf admin summary
```

```
IPv4 Unicast Summary:
```

```
BGP router identifier 192.168.2.1, local AS number 11 vrf-id 2
```

```
BGP table version 3
```

```
RIB entries 5, using 920 bytes of memory
```

```
Peers 3, using 61 KiB of memory
```

```
Peer groups 1, using 64 bytes of memory
```

| Neighbor      | V      | AS    | MsgRcvd | MsgSent | TblVer | InQ | OutQ | Up/Down  | State/ |
|---------------|--------|-------|---------|---------|--------|-----|------|----------|--------|
| ↔ PfxRcd      | PfxSnt | Desc  |         |         |        |     |      |          |        |
| 169.254.0.101 | 4      | 64601 | 10      | 18      | 0      | 0   | 0    | 00:04:30 | ↘      |
| ↔ 1           | 3      | N/A   |         |         |        |     |      |          |        |
| 169.254.0.102 | 4      | 64602 | 10      | 18      | 0      | 0   | 0    | 00:04:30 | ↘      |
| ↔ 1           | 3      | N/A   |         |         |        |     |      |          |        |
| 192.168.2.2   | 4      | 12    | 23      | 23      | 0      | 0   | 0    | 00:18:03 | ↘      |
| ↔ 0           | 3      | N/A   |         |         |        |     |      |          |        |

Note that the above example illustrates peering with eBGP connections with external devices, but also internally with peers on other VRs. Because /32 bit routes have been used to connect to remote veth peers, routes learnt via those peers could not be installed locally because BGP checks against recursivity for eBGP connections. This option has been relaxed, by using `ebgp-connected-route-check` command.

To distinguish the two kinds of peerings, `local-as` option has been used to replace the original AS number with a private AS number to show to peers from veth links. A side effect of this usage is that the AS-PATH list is made up of public and private ASes. To remove private AS and keep only public AS, the following command is used by BGP before sending advertisements to external devices: `as-outbound-update action remove all`.

Below dump gives an extract of RIB of the configured device, and a remote device connected to VRF admin. As you can see, on remote BGP, private ASes have been removed from AS-PATH list.

```
rt1> show bgp vrf admin ipv4
```

```
BGP table version is 3, local router ID is 192.168.2.1, vrf id 2
```

```
Default local pref 100, local AS 11
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
```

```
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```

| Network           | Next Hop      | Metric | LocPrf | Weight | Path       |
|-------------------|---------------|--------|--------|--------|------------|
| *> 10.1.2.0/24    | 169.254.0.101 | 0      |        | 0      | 64601 ?    |
| *> 10.2.2.0/24    | 169.254.0.102 | 0      |        | 0      | 64602 ?    |
| *> 20.1.1.0/24    | 169.254.0.101 |        |        | 0      | 64601 13 i |
| *> 192.168.2.0/24 | 0.0.0.0       | 0      |        | 32768  | ?          |

```
Displayed 3 routes and 3 total paths
```

(continues on next page)

(continued from previous page)

```

rt2> show bgp ipv4
BGP table version is 3, local router ID is 192.168.2.2, vrf id 0
Default local pref 100, local AS 12
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
                i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 10.1.2.0/24      192.168.2.1              0 11 ?
*> 10.2.2.0/24      192.168.2.1              0 11 ?
*> 20.1.1.0/24      192.168.2.1              0 11 13 i
*> 192.168.2.0/24  192.168.2.1              0              0 11 ?

```

An other method to get rid of private ASes would consist in filtering the AS in incoming as-path list on the veth peerings.

```

vrf admin
  routing bgp
    neighbor 192.168.2.2 address-family ipv4-unicast
      del as-outbound-update
    .. .. .
  neighbor-group local
    address-family
      ipv4-unicast
      route-map in route-map-name alteraspath
    ..
  ..
  .. .. .
vrf servicel
  routing bgp
    neighbor 10.1.2.2 address-family ipv4-unicast
      del as-outbound-update
    .. .. .
  neighbor-group local
    address-family
      ipv4-unicast
      route-map in route-map-name alteraspath
    ..
  ..
  .. .. .
vrf service2
  routing bgp

```

(continues on next page)

(continued from previous page)

```
neighbor 10.2.2.2 address-family ipv4-unicast
  del as-outbound-update
  .. .. .
neighbor-group local
  address-family
  ipv4-unicast
  route-map in route-map-name alteraspath
  ..
  ..
  .. .. . . .
routing route-map alteraspath
  seq 100
  policy permit
  match
    peer 169.254.0.100
    ..
  set
    as-path
      exclude 64600
    ..
  ..
  ..
  seq 101
  policy permit
  match
    peer 169.254.0.101
    ..
  set
    as-path
      exclude 64601
    ..
  ..
  ..
  seq 102
  policy permit
  match
    peer 169.254.0.102
    ..
  set
    as-path
      exclude 64602
    ..
  ..
  ..
```

Using a full iBGP network is also an other solution to get rid of AS-PATH list handling. In that case, `nexthop-self` attribute may be used under each peer address family, because the mesh will be incomplete. For instance, the peers behind VRF `admin` would not be connected to peer connected behind VRF `service1`.

Finally, as `veth` is an ethernet medium like many external links, any other routing protocol can be used to transmit routing information to remote endpoints.

## BGP and SNMP (Simple Network Management Protocol)

The BGP4-MIB MIB (Management Information Base) is implemented. Especially, following tables are made available: `bgpPeerTable`, and `bgp4PathAttrTable`. The BGP MIB is only available for the `main vrf`.

---

**Note:** Some flexibility is given for `bgpPeerTable`, as this table will collect all peer entries from all VRs. As a consequence, if the user configures the same peer on 2 separate VRs, only the information of the first retrieved peer will be collected.

---

## BGP for L3VPN

BGP routing protocol is very rich, and permits exchanging more complex information. With the increasing usage of overlay information (widely used in data centers, but also deployed in ISPs), BGP evolves and is able to carry tunneling information through L3VPN. L3VPN stands for the ability to encapsulate IP information, into an other payload. Here, the underlay is an IP packet, and the encapsulation used is MPLS. The overlay IP information is the original IP packet with its payload, coming from one of the mutiple virtual private networks.

## L3VPN overview

L3VPN helps in interconnecting VPNs between several sites, by using a single BGP connection, and at the same time keeping isolation between the different VPNs. L3VPN is based on MPLS technology, and separates the various VPNs connections with MPLS labels handled by BGP. Adding to this, L3VPN offers powerful tools to share (or not) traffic between VPNs by introducing some new RT concepts (see definition below): those tools permit to implement what we call `vrf route leaking` entries. Those routes can also be used without MPLS for defining local importation and exportation policies between VRs. More generally, those concepts apply to EVPN as per *EVPN route target configuration use case*.

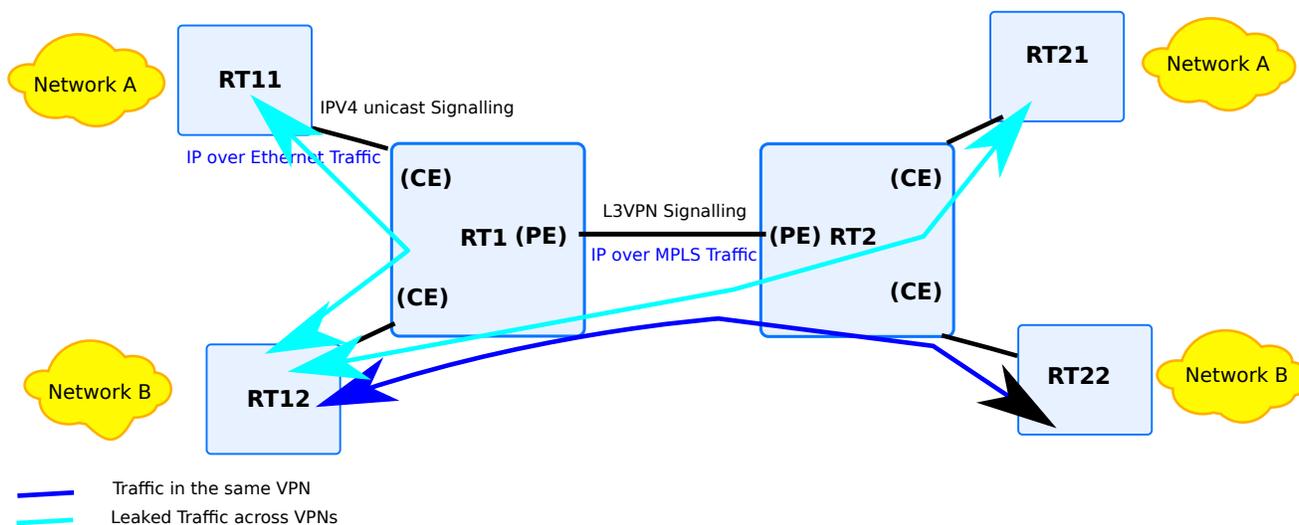


Fig. 6: BGP L3vpn use case example

Above drawing illustrates a setup made up of 2 symmetrical sites. Each site is separated with a PE (Provider Edge) device. In the case the site is a data center, the PE could be replaced with a DC-GW (Data Center Gateway). To simplify, each site is made up of 2 distinct VPNs. L3VPN functionality helps to exchange information about the 2 VPNs, between the 2 sites.

This functionality will subsequently enable data path forwarding. IP Traffic between the 2 PEs (Provider Edges) will be encapsulated into an MPLS label. On each site, traffic between each CE (Customer Equipment) and the PE is standard IP over ethernet traffic.

From the drawing, the following use cases will be more in detail leveraged successively in that document:

- how to interconnect traffic from different VPNs on a same site. This is a specific case from route-leaking. The basic L3VPN commands will be illustrated, as well as the commmands to create CROSS-VRF interfaces across VRF. Note that this kind of method is recommended only if the user has interest in interconnecting also VPNs traffic between sites. If this is not the case, an other method to establish cross-VRF routes is recommended at following link: *BGP use case for cross-vrf routes*.
- how to interconnect traffic from a same VPN between sites. This use case will generalise usage of CROSS-VRF interfaces with default VRF, as well as explaining how to configure backbone so as to carry MPLS labels.
- how to interconnect traffic from different VPNs between sites This use case will generalise the L3VPN use case in an MPLS based framework.

## L3VPN terminology

It is important to understand some L3VPN terminology. In this paragraph we will give the most important concepts.

**VPN** This acronym refers here to a routing entity, also called VRF. Creating a VPN consists in creating a BGP instance in a separated VRF. More information in *BGP VRF*.

**Route Distinguisher, RD (Route Distinguisher)** : This attribute is specific for each VPN. This information is exported along with the L3VPN information of the BGP information

**L3VPN** This refers to creating an overlay with IP packets. In this chapter, L3VPN refers to encapsulating IP traffic over MPLS traffic.

**VPNv4 (Virtual Private Network for IPv4), VPNv6 (Virtual Private Network for IPv6)** This refers to **RFC 4364** (<https://tools.ietf.org/html/rfc4364.html>): that defines how BGP implements L3VPN with MPLS

**Route Target, RT:** RT and RD share the same format. A VPN can have 2 list of RTs (Route Targets). One is dedicated for import. This will help BGP to import incoming routing entries that come from a remote BGP entity. There is also a list for export, that is sent to remote BGP entities. Route Target is the key element for sharing information across VPNs.

**VRF route leaking:** This refers to the ability to share a route from a VPN to an other entity. See chapter *BGP VRF route leak*. This ability requires that the VPN that shares a common route, do not have overlapping with the IP provisioning of the networks.

## Configuring locally route leaking between VPN

### BGP configuration

Below configuration illustrates a setup made up with 2 VPNs. As can be seen, there is a BGP instance in each of the 2 VR instances, plus the BGP core instance. The 2 instances are configured so as to create leaking between both VPNs. Actually configuration shows the following:

- the RD is different, indicating that there are 2 distinct VPNs.
- the RT export settings of each VPN is being matched by the RT import settings of the other VPN. This configuration means that route leaking will occur. Practically, export RT of `customer1` is `11:22`, and import RT of `customer2` is `11:22` and `22:44`. `11:22` value matches, this means that `customer2` will import information coming from `customer1`.

```
vrf main
  routing bgp
    as 65500
    ..
    ..
    ..
vrf customer1
  routing bgp
```

(continues on next page)

(continued from previous page)

```

as 65500
address-family ipv4-unicast
  network 192.168.3.0/24
  ..
  l3vpn export route-distinguisher 1:55
  l3vpn export route-target 11:22
  l3vpn export vpn true
  l3vpn import route-target 11:22 route-target 22:44
  l3vpn import vpn true
  ..
  ..
  ..
  ..
vrf customer2
  routing bgp
  as 65500
  address-family ipv4-unicast
    network 192.168.2.0/24
    ..
    l3vpn export route-distinguisher 2:55
    l3vpn export route-target 22:44
    l3vpn export vpn true
    l3vpn import route-target 11:22 route-target 22:44
    l3vpn import vpn true
    ..
    ..
    ..
    ..
    ..

```

When using above configuration, it is mandatory to create BGP core instance. Also, despite RD and RT values could have been the same, it has been deliberately chosen to have distinct values to better understand the mechanisms put in place when dealing with L3VPN importation and exportation. Also, it is common, when a L3VPN setup is put in place between 2 ISPs, that the RD is self to each operator, while the RT will be chosen accordingly by operators.

---

**Note:** Above configuration details how routing information is exchanged, but does not explain in detail how data traffic is sent through. The product design choices opted for strongly isolating traffic across VPNs. To pass traffic across VPNs, it is required to create special interfaces by configuration. Those interfaces will make the connection between VPNs. More information will be given about how to create those interfaces in a separate chapter. Next chapters assume the user is familiar with interfaces used for crossing vrfs.

---

## using IPless Virtual Ethernet interfaces

Using Cross-VRF interfaces to perform vrf route leaking with BGP requires a specific semantic between VRs and interface names. VR naming must meet the requirements of interface naming. Actually, the Cross-VRF interface name chosen must be equal to the target VR the interface is connected to. To illustrate, in order to reach VR `foo` from VR `bar`, an Cross-VRF interface named `foo` has to be created in VR `bar`. Reversely, an Cross-VRF interface named `bar` has to be created in VR `foo`. In this way, the interface `foo` and the interface `bar` will be connected together. The naming convention is not only done to reflect the intent of the interface. It is mandatory to configure it in this way, if one wants to benefit from route leaking across VRs, using Cross-VRF interfaces, and BGP.

From the user point of view, if a packet is emitted in one interface of a source VR, the same packet will be received in the associated veth interface of destination VR. More information about Cross-VRF interfaces can be found in *XVRF Interface types*. In order to have the setup working fine, the below configuration has to be appended to the former configuration of previous chapter (*BGP VRF leak*).

```
vrf customer1
  interface xvrf customer2
    link-interface customer1
    link-vrf customer2
    ..
    ..
  ..
vrf customer2
  interface xvrf customer1
    link-interface customer2
    link-vrf customer1
    ..
    ..
  ..
```

With the above configuration applied, VR route leaking is possible. Subsequently, if BGP peering is done between a CE and the BGP instance of each VR instance, then route importation and exportation occurs. Below output demonstrates that the routes from `customer2` have been imported to `customer1`. The VR route leak are visible with the `@1<` indicating that the route entry is originated from VR `customer2`.

```
rt1> show bgp vrf customer1 ipv4
BGP table version is 20, local router ID is 1.1.1.1, vrf id 2
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete
```

| Network          | Next Hop | Metric | LocPrf | Weight | Path |
|------------------|----------|--------|--------|--------|------|
| *>i10.101.0.0/24 | 1.1.1.2  | 100    | 0      | 0      | i    |
| *>i10.101.1.0/24 | 1.1.1.2  | 100    | 0      | 0      | i    |
| *>i10.101.2.0/24 | 1.1.1.2  | 100    | 0      | 0      | i    |

(continues on next page)

(continued from previous page)

```
[..]
*> 10.201.0.0/24    2.2.2.3@1<          100    0 i
*> 10.201.1.0/24    2.2.2.3@1<          100    0 i
*> 10.201.2.0/24    2.2.2.3@1<          100    0 i
[..]
```

## Routing output

Once those entries selected in the bgp RIB, nothing prevents the installation of those VRs routes in the underlying system. Packets going from a VR to an other VR are using the CROSS-VRF interface created. There is no specific encapsulation, only a route using an interface as gateway. From above example, the following output displays the routing entries available in VRF routing table. As can be seen, the route to reach the remote prefix first reaches the customer2 interface (first line directly connected, customer2). Then, once the other VR reached, the original BGP route of customer2 routes the traffic to the correct destination (via 2.2.2.3 lines). The latter entry is only here for informational purpose. To get information about routes in VR customer2, use the associated show command to dump route entris in the associates VR.

```
rt1> show ipv4-routes vrf customer1
BGP table version is 20, local router ID is 1.1.1.1, vrf id 2
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
                i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
VRF customer1:
B>* 10.101.0.0/24 [200/0] via 1.1.1.2, eth1_0, 00:05:06
B>* 10.101.1.0/24 [200/0] via 1.1.1.2, eth1_0, 00:05:06
B>* 10.101.2.0/24 [200/0] via 1.1.1.2, eth1_0, 00:05:06
[..]
B>* 10.201.0.0/24 [200/0] is directly connected, customer2, 00:05:06
*
* via 2.2.2.3, eth2_0(vrf customer2), 00:05:06
B>* 10.201.1.0/24 [200/0] is directly connected, customer2, 00:05:06
*
* via 2.2.2.3, eth2_0(vrf customer2), 00:05:06
B>* 10.201.2.0/24 [200/0] is directly connected, customer2, 00:05:06
*
* via 2.2.2.3, eth2_0(vrf customer2), 00:05:06
[..]
```

## how to interconnect traffic from a same VPN between sites.

Being able to interconnect between sites using L3VPN technology is now possible. As explained in the introduction of that chapter, the traffic between sites is encapsulated into an MPLS label, negotiated thanks to BGP. More exactly, the `ipv4-vpn` address-family configured in BGP will obtain from remote the label, and the underlay nexthop to use, to reach the remote. That label will be the encapsulation between 2 VPNs, in one direction. The same mechanism will apply in reverse direction. Adding to this, this label will be conveyed by an other framework. Currently, LDP offers that framework by conveying inner BGP labels in an outer MPLS label negotiated by LDP. More information on chapter (*LDP configuration*).

In order to activate L3VPN, use the following command under the main BGP core instance. L3VPN address-family must be configured on the same VRF where the LDP configuration is. Usually, the backbone is the main VR instance.

```
vrf main
  routing bgp
    as 65500
    neighbor 9.9.9.9 remote-as 65512
    neighbor 9.9.9.9 update-source 3.3.3.3
    neighbor 9.9.9.9 address-family ipv4-vpn enabled true
    ..
    ..
    ..
```

Consequently, having an L3VPN peering will trigger importation of L3VPN entries. For instance, the presence of remote VPNs and associated prefixes from peer 9.9.9.9 will trigger prefixes importation in the relevant VRs. Those prefixes, if the remote VPNs match the local VPNs will be imported in the associated VR. So as to permit that importation, the associated Cross-VRF interfaces will be created between the main VR and the relevant VRs. The following configuration illustrates the Cross-VRF interfaces.

```
vrf main
  interface xvrf customer1
    link-interface main
    link-vrf customer1
    ..
    ..
  interface xvrf customer2
    link-interface main
    link-vrf customer2
    ..
    ..
  ..
vrf customer1
  interface xvrf main
    link-interface customer1
    link-vrf main
```

(continues on next page)

(continued from previous page)

```
..
..
..
vrf customer2
  interface xvrf main
    link-interface customer2
    link-vrf main
  ..
  ..
  ..
```

Also, in order for BGP to be able to export its own labels, BGP must be configured so as to rely on its own labels, either automatically, or by choosing its own. Below configuration illustrates the automatic label chosen by VR customer1, while customer2 chooses to hardset its exportation label to 300.

```
vrf customer1
  routing bgp
    as 65500
    address-family ipv4-unicast
      network 192.168.2.0/24
      ..
      l3vpn export label auto
      ..
      ..
    ..
    ..
  ..
vrf customer2
  routing bgp
    as 65500
    address-family ipv4-unicast
      network 192.168.2.0/24
      ..
      l3vpn export label 300
      ..
      ..
    ..
    ..
  ..
```

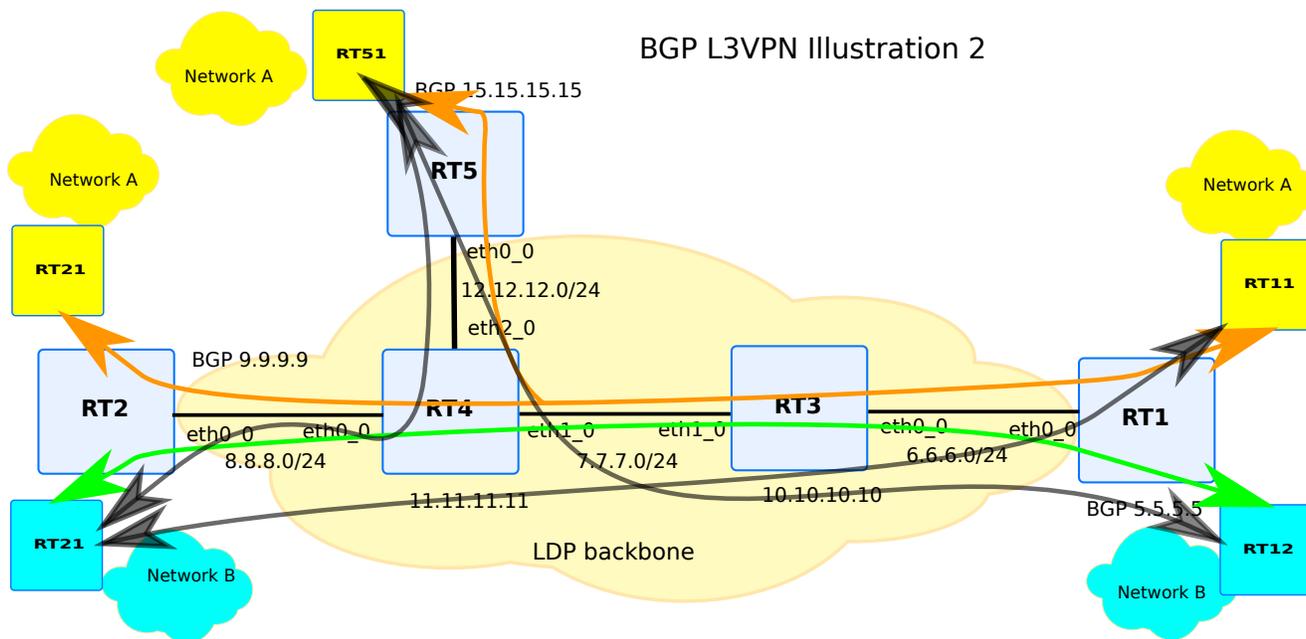


Fig. 7: L3VPN setup using MPLS based framework

Above diagram illustrates a topology made up of an MPLS based backbone, with LSR (Label-Switched Router) devices : `rt3` and `rt4`, and LER (Label Edge Router) devices. Each LER device has a BGP core instance that has `ipv4-vpn` address-family enabled. Next to each BGP instance, a BGP instance is created in each VR. Each VR stands for a private network, either A or B. The color semantic explains the relationship between the private networks on the various LER devices.

As can be seen with the arrows, each private network is geographically separate, but thanks to L3VPN, the private networks act as if there were only 2 specific private networks. The BGP configuration is depicted below. The LDP and OSPF configuration is out of scope of this chapter.

**rt1**

```
vrf main
  routing bgp
    router-id 5.5.5.5
    as 65500
    neighbor 9.9.9.9 remote-as 65500
    neighbor 15.15.15.15 remote-as 65500
    neighbor 9.9.9.9 update-source 5.5.5.5
    neighbor 15.15.15.15 update-source 5.5.5.5
    neighbor 9.9.9.9 address-family ipv4-vpn enabled true
    neighbor 15.15.15.15 address-family ipv4-vpn enabled true
    ..
```

(continues on next page)

(continued from previous page)

```
..
..
vrf customer1
  routing bgp
    router-id 1.1.1.1
    as 65500
    address-family ipv4-unicast
      l3vpn export route-distinguisher 1:55
      l3vpn export route-target 1:55
      l3vpn export vpn true
      l3vpn import route-target 1:55
      l3vpn import vpn true
      l3vpn export label auto
    ..
    ..
    ..
    ..
    ..
vrf customer2
  routing bgp
    router-id 2.2.2.2
    as 65500
    address-family ipv4-unicast
      l3vpn export route-distinguisher 2:55
      l3vpn export route-target 2:55
      l3vpn export vpn true
      l3vpn import route-target 2:55
      l3vpn import vpn true
      l3vpn export label auto
    ..
    ..
    ..
    ..
    ..
```

**rt2**

```
vrf main
  routing bgp
    router-id 9.9.9.9
    as 65500
    neighbor 5.5.5.5 remote-as 65500
```

(continues on next page)

(continued from previous page)

```
neighbor 15.15.15.15 remote-as 65500
neighbor 5.5.5.5 update-source 9.9.9.9
neighbor 15.15.15.15 update-source 9.9.9.9
neighbor 5.5.5.5 address-family ipv4-vpn enabled true
neighbor 15.15.15.15 address-family ipv4-vpn enabled true
..
..
..
vrf customer1
  routing bgp
    router-id 1.1.1.10
    as 65500
    address-family ipv4-unicast
      l3vpn export route-distinguisher 1:55
      l3vpn export route-target 1:55
      l3vpn export vpn true
      l3vpn import route-target 1:55
      l3vpn import vpn true
      l3vpn export label auto
    ..
    ..
  ..
  ..
..
vrf customer2
  routing bgp
    router-id 2.2.2.2
    as 65500
    address-family ipv4-unicast
      l3vpn export route-distinguisher 2:55
      l3vpn export route-target 2:55
      l3vpn export vpn true
      l3vpn import route-target 2:55
      l3vpn import vpn true
      l3vpn export label auto
    ..
    ..
  ..
  ..
..
```

**rt5**

```
vrf main
  routing bgp
    router-id 15.15.15.15
    as 65500
    neighbor 5.5.5.5 remote-as 65500
    neighbor 9.9.9.9 remote-as 65500
    neighbor 5.5.5.5 update-source 15.15.15.15
    neighbor 9.9.9.9 update-source 15.15.15.15
    neighbor 5.5.5.5 address-family ipv4-vpn enabled true
    neighbor 9.9.9.9 address-family ipv4-vpn enabled true
    ..
    ..
  ..
vrf customer1
  routing bgp
    router-id 1.1.1.20
    as 65500
    address-family ipv4-unicast
      l3vpn export route-distinguisher 1:55
      l3vpn export route-target 1:55
      l3vpn export vpn true
      l3vpn import route-target 1:55
      l3vpn import vpn true
      l3vpn export label auto
    ..
    ..
  ..
  ..
  ..
vrf customer2
  routing bgp
    router-id 2.2.2.20
    as 65500
    address-family ipv4-unicast
      l3vpn export route-distinguisher 2:55
      l3vpn export route-target 2:55
      l3vpn export vpn true
      l3vpn import route-target 2:55
      l3vpn import vpn true
      l3vpn export label 300
    ..
    ..
  ..
```

(continues on next page)

(continued from previous page)

```
..
..
```

As can be seen, the network prefixes learnt by each BGP instance are either learnt, by adding extra CE configuration linked to each instance, or imported thanks to L3VPN technology. Below show dumps the vpnv4 entries:

```
rt1> show bgp ipv4 vpn
BGP table version is 27, local router ID is 5.5.5.5, vrf id 0
Default local pref 100, local AS 65500
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1:55
*> 10.101.0.0/24    1.1.1.2@2<        100      0 i
*> 10.101.1.0/24    1.1.1.2@2<        100      0 i
*> 10.101.2.0/24    1.1.1.2@2<        100      0 i
[.]
*>i10.101.11.0/24   9.9.9.9           100      0 i
*>i10.101.12.0/24   9.9.9.9           100      0 i
*>i10.101.13.0/24   9.9.9.9           100      0 i
*>i10.101.14.0/24   9.9.9.9           100      0 i
[.]
Route Distinguisher: 2:55
*> 10.201.0.0/24    2.2.2.3@1<        100      0 i
*> 10.201.1.0/24    2.2.2.3@1<        100      0 i
*> 10.201.2.0/24    2.2.2.3@1<        100      0 i
*> 10.201.3.0/24    2.2.2.3@1<        100      0 i
[.]
*>i10.201.11.0/24   9.9.9.9           100      0 i
*>i10.201.12.0/24   9.9.9.9           100      0 i
*>i10.201.13.0/24   9.9.9.9           100      0 i
*>i10.201.14.0/24   9.9.9.9           100      0 i
[.]
```

As can be seen, all the entries have an underlay nexthop that stands for the nexthop of the remote BGP global instance. If local entries are imported, that nexthop will be used in the route-leak to send traffic to the remote BGP instance. Along with the nexthop, an MPLS label is be sent via BGP, and used on top of the MPLS backbone. MPLS stacking will be performed. The MPLS backbone will handle the LSP (Label-Switched Path) path.

It is worth to be noted too, that the first 3 visible entries are locally exported entries. The nexthop to use 1.1.1.2 is located in the VR customer1. The relationship between the VR name and the VR identifier displayed ( the @2 value) can be done using the following command:

```

rt1> show bgp vrfs
Type  Id      routerId      #PeersCfg  #PeersEstb  Name
      Id      L3-VNI        RouterMAC   Interface
DFLT  0        5.5.5.5      1           1           main
      0        0             00:00:00:00:00:00  unknown
VRF   2        1.1.1.1      1           1           customer1
      0        0             00:00:00:00:00:00  unknown
VRF   1        2.2.2.2      1           1           customer2
      0        0             00:00:00:00:00:00  unknown

```

The label value 80 is the exported value sent to the remote BGP peers. That value is received by remote BGP speaker 9.9.9.9 for instance, along with the undelay network:

```

rt2> show bgp ipv4 vpn
BGP table version is 20, local router ID is 9.9.9.9, vrf id 0
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
                i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1:55
*>i10.101.0.0/24    5.5.5.5           100      0 i
   UN=5.5.5.5 EC{1:55} label=80 type=bgp, subtype=0
*>i10.101.1.0/24    5.5.5.5           100      0 i
   UN=5.5.5.5 EC{1:55} label=80 type=bgp, subtype=0
*>i10.101.2.0/24    5.5.5.5           100      0 i
   UN=5.5.5.5 EC{1:55} label=80 type=bgp, subtype=0
[.]

```

After having checked that L3VPN peering received the correct information, it is possible to check against available entries in the bgp vrf customer1 instance like follows:

```

rt1> show bgp vrf customer1 ipv4
BGP table version is 32, local router ID is 1.1.1.1, vrf id 2
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
                i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*>i10.101.0.0/24    1.1.1.2           100      0 i
*>i10.101.1.0/24    1.1.1.2           100      0 i
*>i10.101.2.0/24    1.1.1.2           100      0 i
[.]

```

(continues on next page)

(continued from previous page)

```
*> 10.101.11.0/24 9.9.9.9@0< 100 0 i
*> 10.101.12.0/24 9.9.9.9@0< 100 0 i
*> 10.101.13.0/24 9.9.9.9@0< 100 0 i
[.]
*> 10.101.22.0/24 15.15.15.15@0< 100 0 i
*> 10.101.23.0/24 15.15.15.15@0< 100 0 i
*> 10.101.24.0/24 15.15.15.15@0< 100 0 i
[.]
```

Like for the previous dump of vpn entries, it is worth to be noted that remote vpn entries have their underlay nexthop visible, but annotated with @0< indicating that this is a VR route leak going to the VPN. Only 1.1.1.2 ip address is a locally reachable IP address. Actually, this is a CE of that instance.

## Routing output

Once those entries selected in the bgp RIB, nothing prevents the installation of those VR routes in the underlying system.

As remind, packets going from a VR to a remote VPN are first mpls encapsulated once. Then, a second encapsulation takes place, as the backbone is MPLS based. Then MPLS does the job to forward the packet to the nexthop marked UN. Reversely, because the BGP vrf instance is at the LER place, the incoming packets are only encapsulated with the negotiated BGP label. So, to go from the backbone VR to the local VR, the packet is being popped its mpls label.

From above example, the following output can be extracted from the VR routing table. As illustrated, the installed route entry performs a double MPLS encapsulation (82/80). The inner value is the negotiated BGP value. The 82 value is an intermediate value that is being swapped by the negotiated MPLS value on the backbone.

```
rt1> show ipv4-routes vrf customer1
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR,
       > - selected route, * - FIB route

VRF customer1:
B>* 10.101.0.0/24 [200/0] via 1.1.1.2, eth1_0, 00:27:24
B>* 10.101.1.0/24 [200/0] via 1.1.1.2, eth1_0, 00:27:24
B>* 10.101.2.0/24 [200/0] via 1.1.1.2, eth1_0, 00:27:24
[.]
B>* 10.101.11.0/24 [200/0] is directly connected, main, label 82/80, 00:26:43
                        via 9.9.9.9(vrf main) (recursive), label 80, 00:26:43
      *
                        via 6.6.6.3, eth0_0(vrf main), label 17, 00:26:43
B>* 10.101.12.0/24 [200/0] is directly connected, main, label 82/80, 00:26:43
```

(continues on next page)

(continued from previous page)

```

        via 9.9.9.9(vrf main) (recursive), label 80, 00:26:43
    *
        via 6.6.6.3, eth0_0(vrf main), label 17, 00:26:43
B>* 10.101.13.0/24 [200/0] is directly connected, main, label 82/80, 00:26:43
        via 9.9.9.9(vrf main) (recursive), label 80, 00:26:43
    *
        via 6.6.6.3, eth0_0(vrf main), label 17, 00:26:43
[. .]
B>* 10.101.22.0/24 [200/0] is directly connected, main, label 84/300, 00:26:40
        via 15.15.15.15(vrf main) (recursive), label 300, 00:26:40
    *
        via 6.6.6.3, eth0_0(vrf main), label 22, 00:26:40
B>* 10.101.23.0/24 [200/0] is directly connected, main, label 84/300, 00:26:40
        via 15.15.15.15(vrf main) (recursive), label 300, 00:26:40
    *
        via 6.6.6.3, eth0_0(vrf main), label 22, 00:26:40
B>* 10.101.24.0/24 [200/0] is directly connected, main, label 84/300, 00:26:40
        via 15.15.15.15(vrf main) (recursive), label 300, 00:26:40
    *
        via 6.6.6.3, eth0_0(vrf main), label 22, 00:26:40
[. .]
    
```

The next command dumps the LFIB (Label Forwarding Information Base) of the backbone. As you can see, the 82 value is replaced by the 17 value, and forwarded thanks to calculated LSP. Actually, this entry stands for the path to rt2. For reverse direction, incoming packets are being popped and redirected to the Cross-VRF interface leading to the VRF ( here customer1).

```

rt1> show mpls table
Inbound          Outbound
Label           Type           Nexthop        Label
-----
[. .]
    80           BGP           customer1
    81           BGP           customer2
    82           BGP           6.6.6.3        17
    83           BGP           6.6.6.3        17
    84           BGP           6.6.6.3        22
    
```

### how to interconnect traffic from different VPNs between sites

By integrating examples of the 2 previous chapters, it becomes possible to perform interconnection between various VPNs between sites. Also, some VR route leaking across VPNs is done. Note that like for previous chapters, configuring BGP only is not enough, since conveying MPLS labels from BGP needs an other MPLS framework to be configured. LDP protocol is the framework that should be used. More information on chapter (:ref: LDP configuration <mpls-ldp-configuration>). This being said, next chapter explores how to simplify, eventually remove the usage of LDP.

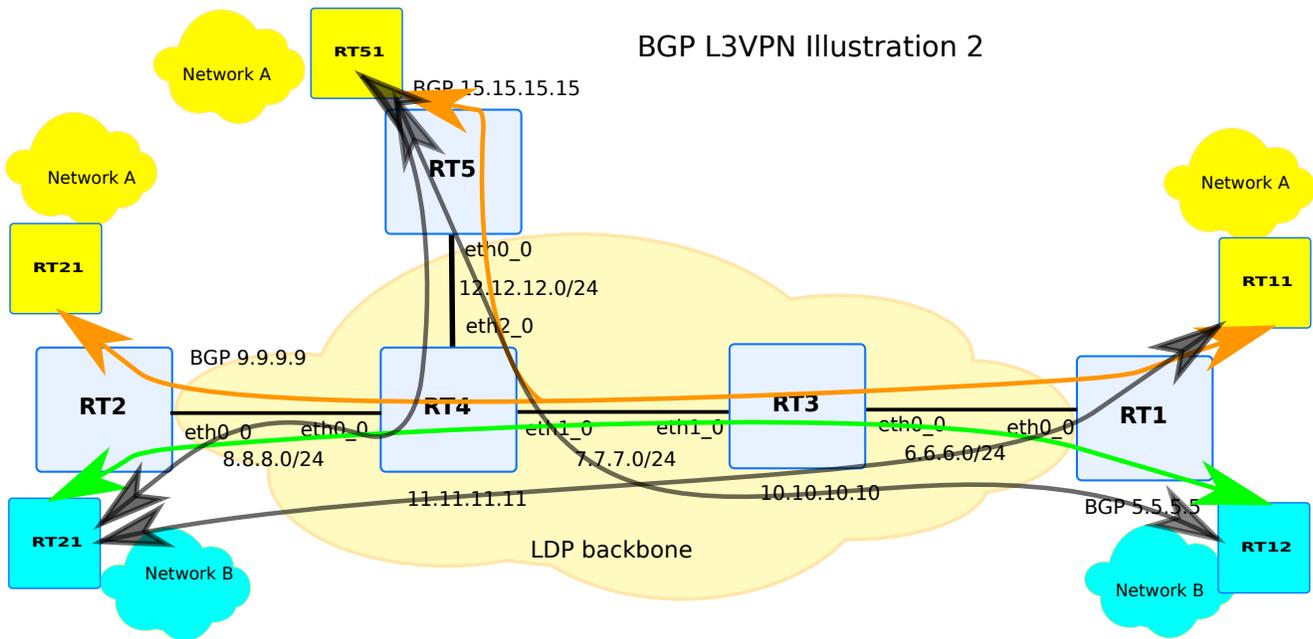


Fig. 8: L3VPN setup using MPLS based framework with VR leaking.

The same topology as for previous chapter is chosen. Black arrows indicate the traffic that will be authorised to pass between different VPNs. For instance, network A and network B can access each other. Note that not all data flow are illustrated in the drawing, but it is also possible to do VR route leak between network A from `rt1` and network B from `rt1`. The L3VPN importation and exportation rules for a defined VPN apply for that VPN, whatever its location is, ie local or remote.

Below configurations are BGP configuration changes. Usually, that kind of configuration can be used, when some resources are shared with other VRs: A video server located on that shared resource, or a management vr that is only able to access to th other VRs.

### rt1

```
vrf main
  routing bgp
    router-id 5.5.5.5
    as 65500
    neighbor 9.9.9.9 remote-as 65500
    neighbor 15.15.15.15 remote-as 65500
    neighbor 9.9.9.9 update-source 5.5.5.5
    neighbor 15.15.15.15 update-source 5.5.5.5
    neighbor 9.9.9.9 address-family ipv4-vpn enabled true
    neighbor 15.15.15.15 address-family ipv4-vpn enabled true
    ..
```

(continues on next page)

(continued from previous page)

```
..
..
vrf customer1
  routing bgp
    router-id 1.1.1.1
    as 65500
    address-family ipv4-unicast
      maximum-path ebgp 4
      maximum-path ibgp 4
      l3vpn export route-distinguisher 1:55
      l3vpn export route-target 1:55
      l3vpn export vpn true
      l3vpn import route-target 1:55 route-target 2:55
      l3vpn import vpn true
      l3vpn export label auto
    ..
    ..
  ..
  ..
..
vrf customer2
  routing bgp
    router-id 2.2.2.2
    as 65500
    address-family ipv4-unicast
      l3vpn export route-distinguisher 2:55
      l3vpn export route-target 2:55
      l3vpn export vpn true
      l3vpn import route-target 1:55 route-target 2:55
      l3vpn import vpn true
      l3vpn export label auto
    ..
    ..
  ..
  ..
```

**rt2**

```
vrf main
  routing bgp
    router-id 9.9.9.9
    as 65500
    neighbor 5.5.5.5 remote-as 65500
    neighbor 15.15.15.15 remote-as 65500
    neighbor 5.5.5.5 update-source 9.9.9.9
    neighbor 15.15.15.15 update-source 9.9.9.9
    neighbor 5.5.5.5 address-family ipv4-vpn enabled true
    neighbor 15.15.15.15 address-family ipv4-vpn enabled true
    ..
  ..
  ..
vrf customer1
  routing bgp
    router-id 1.1.1.10
    as 65500
    address-family ipv4-unicast
      l3vpn export route-distinguisher 1:55
      l3vpn export route-target 1:55
      l3vpn export vpn true
      l3vpn import route-target 1:55 route-target 2:55
      l3vpn import vpn true
      l3vpn export label auto
    ..
  ..
  ..
  ..
  ..
vrf customer2
  routing bgp
    router-id 2.2.2.2
    as 65500
    address-family ipv4-unicast
      l3vpn export route-distinguisher 2:55
      l3vpn export route-target 2:55
      l3vpn export vpn true
      l3vpn import route-target 1:55 route-target 2:55
      l3vpn import vpn true
      l3vpn export label auto
    ..
  ..
  ..
```

(continues on next page)

(continued from previous page)

```
..  
..
```

**rt5**

```
vrf main  
  routing bgp  
    router-id 15.15.15.15  
    as 65500  
    neighbor 5.5.5.5 remote-as 65500  
    neighbor 9.9.9.9 remote-as 65500  
    neighbor 5.5.5.5 update-source 15.15.15.15  
    neighbor 9.9.9.9 update-source 15.15.15.15  
    neighbor 5.5.5.5 address-family ipv4-vpn enabled true  
    neighbor 9.9.9.9 address-family ipv4-vpn enabled true  
    ..  
  ..  
vrf customer1  
  routing bgp  
    router-id 1.1.1.20  
    as 65500  
    address-family ipv4-unicast  
      l3vpn export route-distinguisher 1:55  
      l3vpn export route-target 1:55  
      l3vpn export vpn true  
      l3vpn import route-target 1:55 route-target 2:55  
      l3vpn import vpn true  
      l3vpn export label auto  
    ..  
  ..  
  ..  
  ..  
  ..  
vrf customer2  
  routing bgp  
    router-id 2.2.2.20  
    as 65500  
    address-family ipv4-unicast  
      l3vpn export route-distinguisher 2:55  
      l3vpn export route-target 2:55  
      l3vpn export vpn true  
      l3vpn import route-target 1:55 route-target 2:55
```

(continues on next page)

(continued from previous page)

```

l3vpn import vpn true
l3vpn export label 300
..
..
..
..
..

```

The following show extract on `rt1` dumps the routing entries of A and B located on `rt2`. Two different MPLS labels chosen by BGP of `rt2`, are received by `rt1`, for each network.

```

rt1> show bgp ipv4 vpn
BGP table version is 20, local router ID is 9.9.9.9, vrf id 0
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

```

| Network  | Next Hop | Metric | LocPrf | Weight | Path |
|--|----------|--------|--------|--------|------|
| Route Distinguisher: 2:55                        |          |        |        |        |      |
| [..]   |          |        |        |        |      |
| *>i10.101.11.0/24                                | 9.9.9.9  | 100    | 0      | i      |      |
| UN=9.9.9.9 EC{1:55} label=80 type=bgp, subtype=0 |          |        |        |        |      |
| *>i10.101.12.0/24                                | 9.9.9.9  | 100    | 0      | i      |      |
| UN=9.9.9.9 EC{1:55} label=80 type=bgp, subtype=0 |          |        |        |        |      |
| *>i10.101.13.0/24                                | 9.9.9.9  | 100    | 0      | i      |      |
| UN=9.9.9.9 EC{1:55} label=80 type=bgp, subtype=0 |          |        |        |        |      |
| *>i10.201.11.0/24                                | 9.9.9.9  | 100    | 0      | i      |      |
| UN=9.9.9.9 EC{2:55} label=81 type=bgp, subtype=0 |          |        |        |        |      |
| *>i10.201.12.0/24                                | 9.9.9.9  | 100    | 0      | i      |      |
| UN=9.9.9.9 EC{2:55} label=81 type=bgp, subtype=0 |          |        |        |        |      |
| *>i10.201.13.0/24                                | 9.9.9.9  | 100    | 0      | i      |      |
| UN=9.9.9.9 EC{2:55} label=81 type=bgp, subtype=0 |          |        |        |        |      |

When applied to the underlying system, the encapsulation for packets leaving the `customer1` VR is different, depending if the target prefix belongs to A or B. 2 additional temporary labels are allocated and are swapped as the `show mpls labels` indicate.

```

rt1> show ipv4-routes vrf customer1
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR,
       > - selected route, * - FIB route

```

(continues on next page)

(continued from previous page)

```

VRF customer1:
B>* 10.101.11.0/24 [200/0] is directly connected, main, label 83/80, 00:00:44
    via 9.9.9.9(vrf main) (recursive), label 80, 00:00:44
    *
    via 6.6.6.3, eth0_0(vrf main), label 18, 00:00:44
B>* 10.101.12.0/24 [200/0] is directly connected, main, label 83/80, 00:00:44
    via 9.9.9.9(vrf main) (recursive), label 80, 00:00:44
    *
    via 6.6.6.3, eth0_0(vrf main), label 18, 00:00:44
B>* 10.101.13.0/24 [200/0] is directly connected, main, label 83/80, 00:00:44
    via 9.9.9.9(vrf main) (recursive), label 80, 00:00:44
    *
    via 6.6.6.3, eth0_0(vrf main), label 18, 00:00:44
B>* 10.201.11.0/24 [200/0] is directly connected, main, label 82/81, 00:08:15
    via 9.9.9.9(vrf main) (recursive), label 81, 00:08:15
    *
    via 6.6.6.3, eth0_0(vrf main), label 19, 00:08:15
B>* 10.201.12.0/24 [200/0] is directly connected, main, label 82/81, 00:08:15
    via 9.9.9.9(vrf main) (recursive), label 81, 00:08:15
    *
    via 6.6.6.3, eth0_0(vrf main), label 19, 00:08:15
B>* 10.201.13.0/24 [200/0] is directly connected, main, label 82/81, 00:08:15
    via 9.9.9.9(vrf main) (recursive), label 81, 00:08:15
    *
    via 6.6.6.3, eth0_0(vrf main), label 19, 00:08:15

```

```
rt1> show mpls table
```

| Inbound |      |          | Outbound |
|---------|------|----------|----------|
| Label   | Type | Nexthop  | Label    |
| 80      | BGP  | r1-cust1 |          |
| 81      | BGP  | r1-cust2 |          |
| 82      | BGP  | 6.6.6.3  | 18       |
| 83      | BGP  | 6.6.6.3  | 18       |
| 84      | BGP  | 6.6.6.3  | 18       |
| 85      | BGP  | 6.6.6.3  | 18       |

An additional specificity of the setup is the possibility to import ECMP entries coming from 2 separate locations; here, some 32 bit host routes are retrieved. Some of the entries stand for a server with the same IP, but geographically at 2 different places, namely `rt2` and `rt5`. This kind of scenario can be used for servers that require availability, or where load is split in two, to avoid starvation of the resources of one of the machines.

```
rt1> show bgp ipv4 vpn
```

```

BGP table version is 20, local router ID is 9.9.9.9, vrf id 0
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
                i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

```

(continues on next page)

(continued from previous page)

```

Network          Next Hop          Metric LocPrf Weight Path
*=i10.101.18.10/32 15.15.15.15      100      0 i
  UN=15.15.15.15 EC{1:55} label=300 type=bgp, subtype=0
*>i              9.9.9.9          100      0 i
  UN=9.9.9.9 EC{1:55} label=80 type=bgp, subtype=0
*=i10.101.19.10/32 15.15.15.15      100      0 i
  UN=15.15.15.15 EC{1:55} label=300 type=bgp, subtype=0
*>i              9.9.9.9          100      0 i
  UN=9.9.9.9 EC{1:55} label=80 type=bgp, subtype=0
*=i10.101.20.10/32 15.15.15.15      100      0 i
  UN=15.15.15.15 EC{1:55} label=300 type=bgp, subtype=0
*>i              9.9.9.9          100      0 i

rt1> show ipv4-routes vrf customer1
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR,
       > - selected route, * - FIB route

VRF customer1:
B>* 10.101.18.10/32 [20/0] is directly connected, main, label 19/80, 00:00:09
    * via 9.9.9.9(vrf main) (recursive), label 80, 00:00:09
    * via 6.6.6.3, eth0_0(vrf main), label 20, 00:00:09
    * is directly connected, main, label 22/300, 00:00:09
    * via 15.15.15.15(vrf main) (recursive), label 300, 00:00:09
    * via 6.6.6.3, eth0_0(vrf main), label 21, 00:00:09
B>* 10.101.19.10/32 [20/0] is directly connected, main, label 19/80, 00:00:09
    * via 9.9.9.9(vrf main) (recursive), label 80, 00:00:09
    * via 6.6.6.3, eth0_0(vrf main), label 20, 00:00:09
    * is directly connected, main, label 22/300, 00:00:09
    * via 15.15.15.15(vrf main) (recursive), label 300, 00:00:09
    * via 6.6.6.3, eth0_0(vrf main), label 21, 00:00:09
B>* 10.101.20.10/32 [20/0] is directly connected, main, label 19/80, 00:00:09
    * via 9.9.9.9(vrf main) (recursive), label 80, 00:00:09
    * via 6.6.6.3, eth0_0(vrf main), label 20, 00:00:09
    * is directly connected, main, label 22/300, 00:00:09
    * via 15.15.15.15(vrf main) (recursive), label 300, 00:00:09
    * via 6.6.6.3, eth0_0(vrf main), label 21, 00:00:09

```

## Interconnecting traffic between direct connections

As said in previous interconnection chapters, having an MPLS framework is mandatory to interconnect several VPNs across multiple site. Above examples use LDP. Main reason to use LDP is that MPLS technology must be configured at each hop separating two devices (MPLS is a layer 2 technology). And the border devices should not be aware of the number of hops that separate from remote device. One solution to simplify the configuration and to not rely on external dependencies is to reduce to 1 hop the distance between the two devices that should interconnect. Following examples apply:

- device acting as ASBR (Autonomous System Boundary Router), and directly connecting to remote ASBR. In that case, EBGP is used.
- a GRE tunnel is used to interconnect two devices. As GRE tunnel is a point to point technology, then the traffic flowing across that interface is reduced to 1 hop only, independently of the framework to be crossed.

## Using LDP as framework

Below configuration illustrates the configuration of the two devices interconnected. The configuration does not include the VPN configuration part.

### rt1

```
vrf main
  routing bgp
    router-id 5.5.5.5
    as 65500
    neighbor 9.9.9.9 remote-as 65500
    neighbor 9.9.9.9 update-source 5.5.5.5
    neighbor 9.9.9.9 address-family ipv4-unicast enabled false
    neighbor 9.9.9.9 address-family ipv4-vpn enabled true
    ..
  ..
  interface loopback loop1
    ipv4 address 5.5.5.5/32
    ..
  interface physical eth1
    port pci-b0s4
    ipv4 address 10.1.2.5/24
    ..
  interface gre gre1
    link-interface eth1
    ipv4-address 40.0.0.5/24
    local 10.1.2.5
    remote 10.3.2.9
```

(continues on next page)

(continued from previous page)

```

..
..
routing ospf
  router-id 10.1.2.5
  network 10.1.2.0/24
..
..
routing static
  ipv4-route 9.9.9.9/32 next-hop 40.0.0.9
..
..
routing mpls ldp
  router-id 5.5.5.5
  address-family ipv4
    discovery transport-address 5.5.5.5
  interface gre1
    ..
  ..
..
..
..

```

**rt2**

```

vrf main
  routing bgp
    router-id 9.9.9.9
    as 65500
    neighbor 5.5.5.5 remote-as 65500
    neighbor 5.5.5.5 update-source 9.9.9.9
    neighbor 5.5.5.5 address-family ipv4-unicast enabled false
    neighbor 5.5.5.5 address-family ipv4-vpn enabled true
    ..
  ..
  interface loopback loop1
    ipv4 address 9.9.9.9/32
    ..
    ..
  interface physical eth1
    port pci-b0s4
    ipv4 address 10.3.2.9/24
    ..

```

(continues on next page)

(continued from previous page)

```

..
interface gre gre1
  link-interface eth1
  ipv4 address 40.0.0.9/24
  local 10.3.2.9
  remote 10.1.2.5
..
..
routing ospf
  router-id 10.3.2.9
  network 10.3.2.0/24 area 0
..
..
routing static
  ipv4-route 5.5.5.5/32 next-hop 40.0.0.5
..
..
routing mpls ldp
  router-id 9.9.9.9
  address-family ipv4
    discovery transport-address 9.9.9.9
  interface gre1
    ..
  ..
..
..
..
..
..
..

```

As shown below, an `implicit-null` label has been bound to the route to the next-hop of remote BGP speaker. You can note that `40.0.0.2` is the gateway used to reach remote `9.9.9.9`.

```

rt1> show ipv4-routing
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

```

```

VRF main:
[..]
C>* 5.5.5.5/32 is directly connected, lo, 00:02:21

```

(continues on next page)

(continued from previous page)

```
S>* 9.9.9.9/32 [1/0] via 40.0.0.2, gre1, label implicit-null, 00:00:19
C>* 40.0.0.0/24 is directly connected, gre1, 00:02:21
[.]
```

By adding VPN configuration which is not present on above configuration, one will have two VPN labels negotiated between BGP independently of LDP; 144 is the outgoing label, and 16 is the incoming label.

```
rt1> show bgp ipv4 vpn
BGP table version is 2, local router ID is 5.5.5.5, vrf id 0
Default local pref 100, local AS 65500
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1:1
[.]
*> 172.16.1.0/24    0.0.0.0@1<        0           32768 ?
*>i172.16.2.0/24   9.9.9.9            0    100        0 ?
```

The below `show ipv4-routes` command gives the information on the encapsulation used by a packet going from local vrf `customer1` to remote `customer2`. Outgoing packets leaving `rt1` device are encapsulated with label 144/`implicit-null`. You can note that intermediate step to pass over the vrf border between main and `customer1` happens by appending an intermediate label 17 that is popped when directed to `40.0.0.2` destination IP. The additional `implicit-null` label is chosen since it stands for the label to use when wanting to reach the nexthop of the VPN route. Actually, `9.9.9.9` is reachable via `40.0.0.2` by using `implicit-null` label.

```
rt1> show ipv4-routes vrf customer1
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

VRF r1-cust1:
C>* 172.16.1.0/24 is directly connected, r1-eth0, 00:11:15
B>* 172.16.2.0/24 [200/0] is directly connected, main, label 17/144, 00:09:07
                               via 9.9.9.9(vrf main) (recursive), label 144, 00:09:07
*                               via 40.0.0.2, r1-gre(vrf main), label implicit-null,
↵00:09:07
```

Reversely, incoming packets are received with `16/implicit-null` label. On `rt1`, a switching rule indicates that `16` label is popped, and packet is directed to `xvrf` interface `customer1`.

```
rt1> show mpls table
```

| Inbound<br>Label | Type | Nexthop  | Outbound<br>Label |
|------------------|------|----------|-------------------|
| [..]             |      |          |                   |
| 16               | BGP  | r1-cust1 |                   |
| 17               | BGP  | 40.0.0.2 | implicit-null     |

## Using Static Label configuration

Below configuration illustrates is an alternative to previous configuration. Without LDP, but by relying on static MPLS configuration, it is possible to do convey VPN labels transparently across a backbone, inside GRE tunnels. VPN traffic conveyed will use the static MPLS routes labels. Below static route used label number 3 that stands for ipv4 implicit-null value.

```
rt1
```

```
vrf main
  routing bgp
    router-id 5.5.5.5
    as 65500
    neighbor 9.9.9.9 remote-as 65500
    neighbor 9.9.9.9 update-source 5.5.5.5
    neighbor 9.9.9.9 address-family ipv4-unicast enabled false
    neighbor 9.9.9.9 address-family ipv4-vpn enabled true
    ..
  ..
  interface loopback loop1
    ipv4 address 5.5.5.5/32
    ..
  interface physical eth1
    port pci-b0s4
    ipv4 address 10.1.2.5/24
    ..
  interface gre gre1
    link-interface eth1
    ipv4-address 40.0.0.5/24
    local 10.1.2.5
    remote 10.3.2.9
    ..
  ..
  routing ospf
```

(continues on next page)

(continued from previous page)

```
router-id 10.1.2.5
network 10.1.2.0/24
..
..
routing static
  ipv4-route 9.9.9.9/32 next-hop 40.0.0.9%gre1 label 3
..
..
```

**rt2**

```
vrf main
  routing bgp
    router-id 9.9.9.9
    as 65500
    neighbor 5.5.5.5 remote-as 65500
    neighbor 5.5.5.5 update-source 9.9.9.9
    neighbor 5.5.5.5 address-family ipv4-unicast enabled false
    neighbor 5.5.5.5 address-family ipv4-vpn enabled true
    ..
  ..
  interface loopback loop1
    ipv4 address 9.9.9.9/32
    ..
    ..
  interface physical eth1
    port pci-b0s4
    ipv4 address 10.3.2.9/24
    ..
    ..
  interface gre gre1
    link-interface eth1
    ipv4 address 40.0.0.9/24
    local 10.3.2.9
    remote 10.1.2.5
    ..
    ..
  routing ospf
    router-id 10.3.2.9
    network 10.3.2.0/24 area 0
    ..
    ..
```

(continues on next page)

(continued from previous page)

```

routing static
  ipv4-route 5.5.5.5/32 next-hop 40.0.0.5%gre1 label 3
  ..
  ..

```

Above configuration results in having a label bound to route to reach BGP next-hop, that is to say route via 40.0.0.2. In that case, when resolving route from customer1 VRF, the VPN label would be appended with the static label. The show commands from above example are the same with or without LDP.

## BFD In BGP

With BFD usage in BGP, the failover mechanism is greatly improved by detecting the loss of remote BGP speaker in a few seconds, instead of generally 20/30 seconds. To get more information on BFD, please see *BFD*.

## BFD Configuration And Monitoring In BGP

A BFD peer session context is created, along with BGP peering session. The session inherits from BGP settings. For instance, if `ebgp-multihop` is used, then a BFD session `multi-hop` is created. Also, if `update-source` is used, the `local-address` parameter is set.

```

vrf customer1
  routing bgp
    as 65555
    router-id 192.168.1.1
    neighbor 192.168.1.2 remote-as 65100
    neighbor 192.168.1.2 ebgp-multihop 5
    neighbor 192.168.1.2 update-source 192.168.1.1
    neighbor 192.168.1.2 track bfd

```

Then you can continue the configuration as usual. For timer settings, the default emission and reception settings are set to 300000 microseconds, which may not be what is wished. In that case, it is possible to override default timers, by configuring general timer settings. More information is given in *Configuring general BFD settings*.

```

vrouters> show bfd vrf customer1 sessions
BFD Peers:
peer 192.168.1.2 multihop local-address 192.168.1.1
  ID: 3581662458
  Remote ID: 4190161000
  Status: up
  Uptime: 1 minute(s), 48 second(s)
  Diagnostics: ok
  Remote diagnostics: ok

```

(continues on next page)

(continued from previous page)

```

Local timers:
Receive interval: 600ms
Transmission interval: 600ms
Echo transmission interval: 50ms
Remote timers:
Receive interval: 300ms
Transmission interval: 300ms
Echo transmission interval: 50ms

```

```
vrouter> show bgp vrf customer1 neighbors
```

```

BGP neighbor is 192.168.1.2, remote AS 65100, local AS 65555, external link
Hostname: rt1
  BGP version 4, remote router ID 10.254.254.3, local router ID 10.254.254.1
  BGP state = Established, up for 00:04:37
  Last read 00:00:05, Last write 00:00:05
  Hold time is 24, keepalive interval is 8 seconds
  [...]
  BFD: Type: multi hop
    Detect Multiplier: 3, Min Rx interval: 300, Min Tx interval: 300
    Status: Up, Last

```

## BGP Graceful Restart With BFD

There are cases where the non stop forwarding mechanisms configured in BGP may have to prevent BFD to trigger the neighbouring peer session to go down. BFD provides such feature by embedding in BFD control packet a bit that reflects the relationship between control-plane and dataplane. This bit is called the control bit. By default, that bit is set to 1, and means that if a BFD event happens, then the associated control-plane routing context may go down too.

BGP graceful restart informs remote peer that the local speaker is able to keep BGP routing entries in stale mode, during the non availability of that remote speaker. When leaving, remote BFD peer leaves too. Then, the local BFD triggers a notification to BGP quicker than if the local BGP was detecting that the remote BGP speaker left without saying anything (usually TCP error). When keeping BFD posted with specific BGP constraint, the incoming BFD control packet has the C-BIT unset, which means that the control-plane and dataplane should be independent to each other. Consequently, BGP is notified that the remote BGP speaker went down, but as the incoming C-BIT is unset, the event is ignored, thus letting the BGP graceful restart mechanism taking the hand, and thus keeping the routing entries.

Following configuration should be applied if the control-plane decision should be done independently of the incoming BFD notification. Reversely, that configuration will also unset the C-BIT for outgoing BFD control packets.

```

vrf customer1
  routing bgp

```

(continues on next page)

(continued from previous page)

```

as 65555
router-id 192.168.1.1
graceful-restart
..
neighbor 192.168.1.2 remote-as 65100
neighbor 192.168.1.2 track bfd
neighbor 192.168.1.2 check-control-plane-failure true

```

## BFD Configuration And Monitoring In BGP Using Trackers

It's also possible to configure a BFD or ICMP tracker manually. This enables using the same tracker in different services. The example below uses the same BFD tracker in a BGP neighbor and a static route. If the link becomes unreachable, the BGP neighbor and the static route will be removed from the configuration.

```

tracker bfd my-bfd-tracker
  type multi-hop
  address 192.168.1.2
  source 192.168.1.1
  vrf customer1
  required-receive-interval 600000
  desired-transmission-interval 600000
/
vrf customer1
  routing
    static ipv4-route 192.168.1.0/24 next-hop 192.168.1.2 track my-bfd-tracker
    bgp
      as 65555
      router-id 192.168.1.1
      neighbor 64.120.3.24 remote-as 65100
      neighbor 64.120.3.24 update-source 192.168.1.1
      neighbor 64.120.3.24 track my-bfd-tracker
    /
  /
commit

```

## BGP for EVPN

### Overview

BGP routing protocol is a very rich routing protocols and provide L3VPN and L2VPN features. More information about L3VPN can be read in *BGP L3VPN use case example*. L2VPN stands for the ability to carry layer data (MAC-level frames) over standard encapsulation protocols. More specifically, the underlay is an IP packet with VXLAN header used as encapsulation technique; while the overlay is a layer 2 frame containing MAC information

and IP information. BGP is able to use the benefits of VXLAN tunnels. This permits ISPs to provide network segmentation in VNI (VXLAN Network Identifier) (virtual network identifier in a vxlan header) instead of using VLAN for segmenting the network. Also BGP is well suited to handle IP routing of the underlay. Generally, EVPN sits on PE machines.

EVPN is able to carry IP information, but also MAC information in its signaling protocol. The information is collected from routing tables of each VPN, but also neighboring tables, and bridge tables. Actually, layer 2 connectivity can be obtained thanks to information contained in a virtual bridge attached to the vxlan interface.

EVPN is also able to handle BUM (Broadcast Unknown-Unicast and Multicast) traffic, like if it was a local switch. As vxlan interface is a bridge port with possibly multiple tunnel endpoint entries on the same port, outgoing BUM packets are duplicated and sent to various endpoints.

Also, EVPN uses the same semantic as for L3VPN by handling RTs in extended communities, and using RD in NLRI prefixes. This permits easier interconnection between sites.

Initially, the EVPN standard has first been declined with MPLS underlay (with **RFC 7432** (<https://tools.ietf.org/html/rfc7432.html>)). The main features of EVPN have been proposed, leveraging layer 2 connectivity. Then, the technology evolved, with the increasing usage of overlay technology in data centers. Inter Subnet Forwarding concept has been proposed. This routing mode has been introduced in EVPN, and permits routing overlay information between different sites, similar to what L3VPN technology does with MPLS. Practically, once the routing information exchanged, a bridge interface is used at each side, where packets are routed. Then the MAC layer used to forge overlay packets are the MAC addresses of the bridge interfaces.

## EVPN terminology

**Ethernet VPN, EVPN:** This refers to creating an overlay with layer 2 frames. In this chapter, it refers to encapsulating IP traffic over VXLAN tunnel. In multi protocol BGP, EVPN refers to a specific address family with AFI identifier set to 25 and SAFI identifier set to 70.

**Route Distinguisher, RD:** This attribute is specific for each VPN. This information is exported along with the EVPN information of the BGP information. By configuration, a VPN is often associated to a VNI.

**Route Target, RT:** RT and RD share the same format. An EVPN can have 2 list of RTs. One is dedicated for import. This will help importing MAC entries to the appropriate VPN if it deals with route type 2 entries, or IP to the BGP instance associated to the VPN of the instance, if it deals with route type 5 entries. The other one is dedicated for export, and is proposed in the BGP update message where either RT2 (EVPN Route Type 5) or RT2 prefixes are encoded and shared with other VPNs.

**Route type 2, RT2:** This prefix refers to the list of attributes used to define a MAC entry in the EVPN concept. It is made up of a RD, a MAC address, an optional associated IP address, an EVI (Ethernet Virtual Identifier), and an ESI (Ethernet Segment Identifier). The two last concepts are not used in below examples, but are respectively related to broadcast domain separation, and multi-homing. In VXLAN topology, the VNI comes along with the prefix and is encoded in the MPLS label field of the prefix (because initially, EVPN protocol has been made for MPLS). So, that value is not an MPLS value, and can be decoded without looking at BOS (Bottom Of Stack) value like for MPLS.

**Route type 5, RT2:** This prefix refers to the list of attributes used to define an IP entry located behind a virtual switch in routing mode. It is made up of a RD, an IP address, an EVI, and an ESI. The two last concepts are

not used in below examples, but are respectively related to broadcast domain separation, and multi-homing. In VXLAN topology, the VNI comes along with the prefix and is encoded in the MPLS label field. In the EVPN context, that value is not an MPLS value, and can be decoded without looking at BOS value like for MPLS.

**Route type 3, RT3 (EVPN Route Type 3):** This prefix stands for the inclusive multicast ethernet tag route, and is used to signify that a sub network defined by its RD accept BUM traffic. The prefix comes along with the tunnel end-point; that means that BUM traffic can be sent to that tunnel endpoint.

## Configuring EVPN

### Principles of configuration

The following chapters enter more in detail on how to route or bridge traffic into VXLAN tunnels. The BGP services are differently configured, whether routing mode or bridging mode is used. Let us begin with the bridge and vxlan interfaces.

### bridge and vxlan intrerfaces

To be able to perform EVPN, the core technology relies on a VXLAN interface bridged with a bridge interface. The VXLAN interface link-interface must be on the same VRF where the backbone is, that is to sat the main vrf. Also you can note that you have to colocate both VXLAN and bridge interfaces on the same VRF. There is no other restriction regarding in which VRF both interfaces should be. Regarding the VXLAN interface, VNI value will be configured. The destination IP address of VXLAN interface does not need to be configured, as BGP will configure its own destination IP on the underlay. The configured VXLAN and bridge interfaces will be used by BGP to discover which VNI is present on the device, and which information to send to remote peers.

### EVPN service

In order to activate EVPN, use the following command under the main BGP core instance. `l2vpn-evpn` address-family must be configured. Here, the main BGP core instance will play the backbone role.

```
vrf main
  routing
    bgp
      as 65500
      address-family
        l2vpn-evpn
          advertise-all-vni true
        ..
      ..
```

The `advertise-all-vni` keyword will trigger local discovery of all vxlan and bridge interfaces available, so that BGP will retrieve VNI and use that information to send to remote peers. It is to be noted that the discovery takes into account all `vr-s` instances. So basically, whatever where the VRF is, all VXLAN interfaces will be learnt.

For bridging mode, configuring the BGP main instance is enough. In routing mode, it is usual to configure overlay networking information in separate VRs. For that, if a VRF is dedicated to routing network into a VXLAN interface, then an additional BGP instance attached to the new VRF instance will need to be created in order to perform routing mode. Also, the VRF will be mapped to the VNI by configuration. Subsequently, extra configuration can be done under each BGP instance, directly under the `address-family 12vpn-evpn address-family`.

Below figure illustrates what does routing mode and bridging mode means. As can be seen, two pe devices are interconnected with EVPN. On each device, a VXLAN interface and a bridged interface are linked together, as well as an ethernet port connected to local host devices. The VRF where both bridge and vxlan interfaces sit does not matter, provided that the link information of the VXLAN interface is on the main VRF.

Two data flows are illustrated. The scheme reuses the same VXLAN interface, but practically, it is necessary to create a VXLAN interface for each kind of connectivity. On the one hand, the blue one stands for layer 2 connectivity. Data traffic is bridged on pe devices, that is to say that traffic is bridged from ethernet port to vxlan port, where traffic is encapsulated into vxlan header and transmitted to remote pe. From `hostA` to `hostB`, traffic is full layer 2. On the other hand, the green one stands for layer 3 connectivity. This flow connects two networks together, namely `network B` and `network C`. All happens as if the traffic from `network B` was redirected to gateway in `rt2`, except that the gateway of `rt2` is the remote bridged interface. The forged packet inside VXLAN packet will then be made of the MAC addresses of the two bridged interfaces.

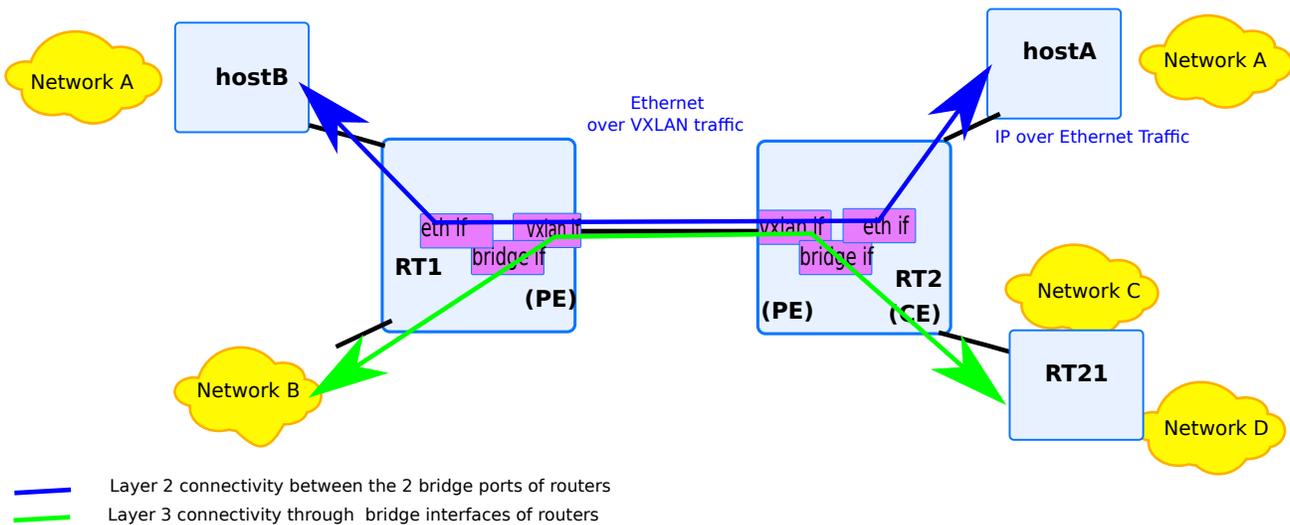


Fig. 9: BGP evpn use case example

## Basic Configuration

EVPN uses a new address-family with AFI identifier set to 25 and SAFI identifier set to 70. Configuring EVPN goes along with the configuration of a bridge interface bridged with a VXLAN interface.

Here below is an example on how to configure a sample BGP configuration with EVPN enabled. As illustrated below, the configuration must include the presence of both vxlan interface and bridge interface.

```
vrf main
  routing
    bgp
      router-id 10.125.0.1
      as 65500
      address-family
        l2vpn-evpn
          advertise-all-vni true
          vni 11
            advertise-default-gw true
          export
            route-distinguisher 65500:11
          ..
        ..
      neighbor 10.125.0.3
        remote-as 65500
        address-family l2vpn-evpn enabled true
      /
vrf custom1
  interface
    physical eth0
      ipv4 address 10.125.0.1/24
      port pci-b0s5
    ..
  vxlan vxl11
    vni 11
      local 10.125.0.1
      link-vrf main
      learning true
      link-interface eth0
      link-vrf main
    ..
  ..
  interface
    bridge br11
```

(continues on next page)

(continued from previous page)

```

link-interface vxl11
..
..

```

There is a single global command to enable the EVPN control plane on a VTEP (VXLAN Tunnel End Point) called `advertise-all-vni`. This will cause the router to learn about all VNIs (VXLAN Network Identifiers) locally present on the system and the MACs and neighbors (ARP and ND (Neighbor Discovery)) that pertain to such VNIs and advertise the corresponding information using EVPN procedures to all BGP peers with whom the EVPN address-family has been negotiated. It will also cause any EVPN routes learnt from BGP peers to be installed into the appropriate local VNIs. Received EVPN type-3 routes will translate into the list of remote VTEPs (VXLAN Tunnel End Points) that participate in a particular VNI and received EVPN type-2 routes will get installed as MAC and neighbor entries pertaining to a specific VNI.

It is to be noted that the BGP core instance is used to carry EVPN information, while the other VRs are optionally used to carry the overlay information, be it layer 2 and/or layer 3 information. The mapping between overlay and underlay is based with VNI presence. That implies that the VRs configuration is optional, and for instance, the configuration of a bridge and a virtual interface can be done in the main instance. all traffic that will go through that bridge interface will be subsequently encapsulated, and signaling information will be detected and transmitted within the associated VNI.

Note that we don't define the remote endpoint of the vxlan interface, as the BGP peer defines it, using the VXLAN interface. That vxlan interface link interface is the same interface where underlay traffic goes through.

To get information about the VXLAN interfaces detected, classified per VNI, the following command can be used to dump the contexts. If the VXLAN interfaces have not been detected, then that implies that a misconfiguration occurred, for instance, if the VXLAN interface has not been bridged. Below example shows that the vxl11 has been detected on vrf custom1, and that a certain number of entries have been learnt, either via bgp learning, or by locally listening for ARP/MAC information ( see MACs and MAC information). The first mac information learnt is the MAC address of the bridged interface.

```

rt1> show evpn vni all
VNI          Type VxLAN IF          # MACs  # ARPs  # Remote VTEPs  Tenant VRF
11           L2   vxl11             1       1       1               custom1

rt1> show evpn vni 11
VNI: 11
Type: L2
Tenant VRF: custom1
VxLAN interface: vxl11
VxLAN ifIndex: 6
Local VTEP IP: 10.125.0.1
Mcast group: 0.0.0.0
Remote VTEPs for this VNI:
 10.125.0.3 flood: HER
Number of MACs (local and remote) known for this VNI: 1

```

(continues on next page)

(continued from previous page)

```

Number of ARPs (IPv4 and IPv6, local and remote) known for this VNI: 3
Advertise-gw-macip: Yes

rt1> show evpn arp-cache vni 11
Number of ARPs (local and remote) known for this VNI: 1
IP                Type   State   MAC                Remote VTEP          Seq #
↪ 's
fe80::5814:dbff:feba:c854 local active f2:de:f1:b6:4e:59          0/0

```

To get information about the BGP information exchanged, the following command can be used. Each entry stands for an EVPN route entry. The first number stands for the kind of information shared, ie the route type. For instance, 2 stands for MAC-level information shared (RT2, while 3 (RT3) stands means that this tunnel endpoint is authorized to exchange BUM traffic. The tunnel endpoint can be seen here with the nexthop information. As depicted below, 10.125.0.3 is the tunnel endpoint.

```

rt1> show bgp l2vpn evpn
BGP table version is 4, local router ID is 10.125.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-2 prefix: [2]:[EthTag]:[MACLen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]

   Network                Next Hop                Metric LocPrf Weight Path
Route Distinguisher: 65500:11
*>i [2]:[0]:[48]:[6e:1c:50:ab:2b:00]:[128]:[fe80::acf9:13ff:fe0d:607a]
      10.125.0.3                100      0 i
      RT:65500:11 ET:8 Default Gateway ND:Router Flag
*> [2]:[0]:[48]:[f2:de:f1:b6:4e:59]:[128]:[fe80::5814:dbff:feba:c854]
      10.125.0.1                32768 i
      ET:8 RT:65500:11 Default Gateway ND:Router Flag
*> [3]:[0]:[32]:[10.125.0.1]
      10.125.0.1                32768 i
      ET:8 RT:65500:11
*>i [3]:[0]:[32]:[10.125.0.3]
      10.125.0.3                100      0 i
      RT:65500:11 ET:8

Displayed 4 out of 4 total prefixes

```

It is also possible to do some variants of the call by filtering based on the route type.

```
rt1> show bgp l2vpn evpn route type multicast
```

(continues on next page)

(continued from previous page)

```

BGP table version is 4, local router ID is 10.125.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-2 prefix: [2]:[EthTag]:[MACLen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]

```

| Network                       | Next Hop         | Metric | LocPrf | Weight | Path |
|-------------------------------|------------------|--------|--------|--------|------|
| Extended Community            |                  |        |        |        |      |
| Route Distinguisher: 65500:11 |                  |        |        |        |      |
| *> [3]:[0]:[32]:[10.125.0.1]  | 10.125.0.1       |        |        | 32768  | i    |
|                               | ET:8 RT:65500:11 |        |        |        |      |
| *>i[3]:[0]:[32]:[10.125.0.3]  | 10.125.0.3       | 100    |        | 0      | i    |
|                               | RT:65500:11 ET:8 |        |        |        |      |

Displayed 2 prefixes (2 paths) (of requested type)

Also, it is possible to get some details

```

rt1> show bgp l2vpn evpn route detail
[.]
Route Distinguisher: 65500:11
BGP routing table entry for 65500:11:[2]:[0]:[48]:[6e:1c:50:ab:2b:00]:[128]:[fe80::acf9:13ff:fe0d:607a]
Paths: (1 available, best #1)
Not advertised to any peer
Route [2]:[0]:[48]:[6e:1c:50:ab:2b:00]:[128]:[fe80::acf9:13ff:fe0d:607a] VNI 11
Local
  10.125.0.3 from 10.125.0.3 (10.125.0.3)
  Origin IGP, localpref 100, valid, internal, best (First path received)
  Extended Community: RT:65500:11 ET:8 Default Gateway ND:Router Flag
  Last update: Thu Aug 19 14:05:07 2021
[.]
BGP routing table entry for 65500:11:[3]:[0]:[32]:[10.125.0.3]
Paths: (1 available, best #1)
Not advertised to any peer
Local
  10.125.0.3 from 10.125.0.3 (10.125.0.3)
  Origin IGP, localpref 100, valid, internal, best (First path received)
  Extended Community: RT:65500:11 ET:8
  Last update: Thu Aug 19 14:05:07 2021

```

(continues on next page)

(continued from previous page)

```
PMSI Tunnel Type: Ingress Replication, label: 11
[..]
```

## Route Reflector configuration

It is possible to create a route reflector configuration, then it is not necessary to call `advertise-all-vni` keyword. Enabling `l2vpn-evpn` address-family is enough. EVPN routes received from a BGP peer are accepted and maintained in the global EVPN routing table.

## Flooding Mode

VXLAN interfaces handled by BGP can be configured with or without acceptance of BUM traffic. By default, head-end replication is used; that implies that all BUM traffic entering to the bridge is sent to the other ports of the bridge. That includes the VXLAN interface. For instance, ARP packets are transmitted through the VXLAN interface. To block that traffic, it is possible to disable flooding, and configure BGP so as to forbid BUM traffic. To inform remote peer that flooding is accepted, RT3 messages are sent. This message indicates that for a given network defined by the RD, BUM traffic is accepted. flooding can be configured as follows:

```
vrf main
  routing bgp
    address-family l2vpn-evpn
      flooding disabled
    ..
  ..
  ..
  ..
```

It is possible to reenable flooding by using following command. Consequently, RT3 updates will be propagated.

```
vrf main
  routing bgp
    address-family l2vpn-evpn
      flooding head-end-replication
    ..
  ..
  ..
  ..
```

## Route Target Configuration

More information about route targets is given in *route leaking use case*. In L3VPN chapter, it was seen that RD and RTs were used to import and export routes to different VPNs. Here, the same concepts are used, and are applied to all kind of EVPN route types routes. This includes IP routes, but also MAC entries too. Also, the VPN concept is mapped to VNI concept. This is why we refer to VNI.

In addition to the above essential steps, the RD and RTs can be configured for a VNI. By default, RD is automatically derived by using IP4B:NN format, where IP4B stands for the IP address of the router-id used in BGP, and a unique 16 bit field identifier. RTs values is defined as AS:VNI, where AS stands for the AS value of the BGP instance, and VNI is the virtual network identifier of the VXLAN interface. It is possible to redefine the RD value by using another semantic. Below configurations partially reuse the L2 vni configuration, and append a new L3 vni. As described, configuring RD and RTs for L2 (under vni node), and L3 node (under l2vpn-epvn node of separate BGP instance) differs:

```
vrf main
  routing bgp
    router-id 10.125.0.1
    as 65500
    address-family l2vpn-epvn
      vni 11
        export route-distinguisher 65000:11
/ vrf custom2
  routing bgp
    router-id 10.125.1.1
    as 65500
    address-family l2vpn-epvn
      export route-distinguisher 65000:12
    ..
  ..
  ..
```

It is also possible to override route target values, by using following command. Here, the VNI is used for encoding.

```
vrf main
  routing bgp
    router-id 10.125.0.1
    as 65500
    address-family l2vpn-epvn
      advertise-all-vni true
      vni 11
        export route-target 65000:11
        import route-target 65000:11
/ vrf custom2
  routing bgp
```

(continues on next page)

(continued from previous page)

```

router-id 10.125.1.1
as 65500
address-family l2vpn-evpn
    export route-target 65000:12
    import route-target 65000:12

```

**RFC 8365** (<https://tools.ietf.org/html/rfc8365.html>) explains how RT auto derivation should be done in section 5.1.2.1. The lowest 4 bytes of the AA:NNNN are redefined. The new format is made up of the value 1 in the first 3-bit field (standing for VXLAN encapsulation), and the VNI value. This encoding is needed for proper interoperability with RT auto-derivation in Junos. To configure this format automatically, use the following command:

```

vrf main
  routing bgp
    router-id 10.125.0.1
    as 65500
    address-family
      l2vpn-evpn
        auto-route-target rfc8365
/ vrf custom2
  routing bgp
    router-id 10.125.1.1
    as 65500
    address-family l2vpn-evpn
      auto-route-target rfc8365

```

The RD and RT configuration can be checked, against each VNI discovered.

```

rt1> show bgp l2vpn evpn vni 11
VNI: 11 (known to the kernel)
Type: L2
Tenant-Vrf: custom1
RD: 10.125.0.3:2
Originator IP: 10.125.0.3
Mcast group: 0.0.0.0
Advertise-gw-macip : Enabled
Advertise-svi-macip : Disabled
Import Route Target:
  65500:268435467
Export Route Target:
  65500:268435467

```

As said in introduction chapter, EVPN can be used to carry either L2VPN information or L3VPN information. The next chapters respectively discuss how to use EVPN as a L3VPN technology, then as a L2VPN technology.

## Inter Subnet Forwarding

Inter Subnet Forwarding is the ability to use a virtual bridge to route information on that bridge. BGP exchanges this information by using RT2 messages. As underlay tunnel carries also MAC-level information, the source and destination MAC addresses used ( and transmitted via BGP) are the MAC addresses of the bridge interface attached. Like for L3VPN, IP prefixes can be assigned to a specific VR, and the bridged interfaces will act as both tunnel endpoint and remote gateway to join a separate remote IP network.

To configure routing, by using a VXLAN interface, its VNI must be configured as an L3VPN vni. By default, each VNI presence detection is seen as a EVPN one. You have to explicitly mention the VNI as a layer 3 VNI.

```
vrf custom1
  routing l3-vni 11
  interface vxlan vxlan-101 vni 11
  routing bgp as 65500
```

Subsequently, the VNI layer information is propagated in the system.

```
rt1> show evpn vni all
```

| VNI | Type | VxLAN IF | # MACs | # ARPs | # Remote VTEPs | Tenant VRF |
|-----|------|----------|--------|--------|----------------|------------|
| 11  | L3   | vx111    | 1      | 1      | n/a            | custom1    |

The remaining configuration is same as the one presented in the first chapter, ie both a vxlan and slave to a bridge interface in the same VR. To bring clarity, the whole configuration is reused, and is based on ref:BGP EVPN use case example <routing-bgp-evpn-drawing>, where rt1 and rt2 devices configuration are exposed.

### rt1

```
vrf main
  routing
    bgp
      as 65500
      router-id 10.125.0.1
      address-family
        l2vpn-evpn
          advertise-all-vni true
        ..
      ..
  neighbor 10.125.0.2
    remote-as 65500
    address-family
      l2vpn-evpn
    ..
  ..
```

(continues on next page)

(continued from previous page)

```
    ..
  ..
interface physical eth0
  port pci-b0s4
  mtu 1550
  ipv4 address 10.125.0.1/24
  ..
  ..
..
vrf custom1
  interface physical eth1
    ipv4 address 10.51.0.1/24
    port pci-b0s6
    ..
  ..
interface vxlan vxl11
  vni 11
  local 10.125.0.1
  link-interface eth0
  link-vrf main
  ..
..
interface bridge br11
  link-interface vxl11
  ipv4 address 10.50.0.1/24
  ..
..
routing
  l3-vni 11
  bgp
    as 65500
    address-family l2vpn-evpn
      advertise ipv4-unicast
      auto-route-target rfc8365
      export route-distinguisher 65500:11
    ..
    ..
    address-family ipv4-unicast
      redistribute connected
    ..
  ..
  ..
  ..
  ..
  ..
  ..
```

**rt2**

```
vrf main
  routing
    bgp
      router-id 10.125.0.2
      as 65500
      address-family
        l2vpn-evpn
          advertise-all-vni true
          ..
        ..
      neighbor 10.125.0.1
        remote-as 65500
        address-family
          l2vpn-evpn
          ..
        ..
      ..
    ..
  ..
interface physical eth0
  port pci-b0s4
  mtu 1550
  ipv4 address 10.125.0.2/24
  ..
  ..
..
vrf custom1
  interface physical eth1
    ipv4 address 10.52.0.2/24
    port pci-b0s6
    ..
  ..
  interface vxlan vxl11
    vni 11
    local 10.125.0.2
    link-interface eth0
    link-vrf main
    ..
  ..
  interface bridge br11
    link-interface vxl11
    ipv4 address 10.50.0.2/24
```

(continues on next page)

(continued from previous page)

```

..
..
routing
  l3-vni 11
  bgp
    as 65500
    address-family l2vpn-evpn
      advertise ipv4-unicast
      auto-route-target rfc8365
      export route-distinguisher 65500:11
    ..
  ..
  address-family ipv4-unicast
    redistribute connected
  ..
  ..
  ..
  ..
  ..
  ..

```

To propagate IP information in the L2VPN, a second BGP instance is created in the VR. That instance will explicitly tell to advertise IP information to the main core instance. That second BGP instance redistributes sub networks from eth1 interface, as depicted in below command:

```

rt1> show bgp l2vpn evpn
BGP table version is 12, local router ID is 10.125.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-2 prefix: [2]:[EthTag]:[MACLen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]

   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 65500:11
*> [5]:[0]:[24]:[10.51.0.0]
      10.125.0.1          0          32768 ?
      ET:8 RT:65500:268435467 Rmac:42:f7:18:0a:8e:72
*>i [5]:[0]:[24]:[10.52.0.0]
      10.125.0.2          0   100      0 ?
      RT:65500:268435467 ET:8 Rmac:9a:2a:20:2f:07:69
*> [5]:[0]:[32]:[10.50.0.1]
      10.125.0.1          0          32768 ?
      ET:8 RT:65500:268435467 Rmac:42:f7:18:0a:8e:72

```

(continues on next page)

(continued from previous page)

```
*>i[5]:[0]:[32]:[10.50.0.2]
      10.125.0.2           0    100    0 ?
      RT:65500:268435467 ET:8 Rmac:9a:2a:20:2f:07:69
```

Displayed 4 out of 4 total prefixes

The below commands show that the imported route entries from remote peers have been accordingly set to the VR. Also, the EVPN entries transmitted in the core instance are dumped too.

```
rt1> show bgp vrf custom1 ipv4
BGP table version is 10, local router ID is 10.125.0.1, vrf id 11
Default local pref 100, local AS 65500
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
                i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete
```

| Network         | Next Hop    | Metric | LocPrf | Weight | Path |
|-----------------|-------------|--------|--------|--------|------|
| *> 10.50.0.1/32 | 0.0.0.0     | 0      |        | 32768  | ?    |
| *>i10.50.0.2/32 | 10.125.0.2< | 0      | 100    | 0      | ?    |
| *> 10.51.0.0/24 | 0.0.0.0     | 0      |        | 32768  | ?    |
| *>i10.52.0.0/24 | 10.125.0.2< | 0      | 100    | 0      | ?    |

Displayed 4 routes and 4 total paths

That second BGP instance can also be used to connect with remote CE. This permits extending the L3 connectivity behind each PE devices. The propagated routes transmitted to the CE will use the PE as next-hop to reach that subnetwork.

To unconfigure a layer 3 VNI and its associated BGP core instance, use the following command to remove both configuration. Optionally, the bridge and VXLAN interface configuration can be removed:

```
vrf custom1
  del routing l3-vni
  del routing bgp
  del interface vxlan vxl11
  del interface bridge br11
```

## Bridge Configuration

Bridging permits to aggregate layer 2 networks into a single one, by bridging each network with VXLAN interface. Like for routing, a bridge and a vxlan interface are needed, and need to be bridged, so that BGP populates its VNI list.

Subsequently, the VNI layer information is propagated in the system. BGP uses the VNI information to extract the bridge neighboring information contained to transmit it by using RT2 entries. Based on *BGP EVPN use case example*, `rt1` and `rt2` devices configuration are exposed below:

### rt1

```
vrf main
  routing
    bgp
      router-id 10.125.0.1
      as 65500
      address-family
        l2vpn-evpn
          advertise-all-vni true
          auto-route-target rfc8365
          vni 11
            advertise-default-gw true
            export route-distinguisher 65500:11
          ..
        ..
      neighbor 10.125.0.2
        remote-as 65500
        address-family
          l2vpn-evpn
            ..
          ..
        ..
      ..
  interface physical eth0
    port pci-b0s4
    mtu 1550
    ipv4 address 10.125.0.1/24
    ..
    ..
  ..
vrf custom1
```

(continues on next page)

(continued from previous page)

```

interface physical eth1
  port pci-b0s6
  ..
..
interface vxlan vxl11
  vni 11
  local 10.125.0.1
  link-interface eth0
  link-vrf main
  ..
..
interface bridge br11
  link-interface vxl11
  link-interface eth1
  ipv4 address 10.50.0.1/24
  ..
..

```

**rt2**

```

vrf main
  routing
    bgp
      router-id 10.125.0.2
      as 65500
      address-family
        l2vpn-evpn
          advertise-all-vni true
          auto-route-target rfc8365
          vni 11
            advertise-default-gw true
            export route-distinguisher 65500:11
          ..
        ..
      ..
    neighbor 10.125.0.1
      remote-as 65500
      address-family
        l2vpn-evpn
          ..
        ..
      ..

```

(continues on next page)

(continued from previous page)

```

    ..
    ..
interface physical eth0
  port pci-b0s4
  mtu 1550
  ipv4 address 10.125.0.2/24
  ..
  ..
..
vrf custom1
  interface physical eth1
    port pci-b0s6
    ..
  ..
interface vxlan vxl11
  vni 11
  local 10.125.0.2
  link-interface eth0
  link-vrf main
  ..
  ..
interface bridge br11
  link-interface vxl11
  link-interface eth1
  ipv4 address 10.50.0.2/24
  ..
  ..

```

A summary of the discovered VXLAN interfaces can be seen with following command:

```

rt1> show evpn vni 11
VNI: 11
Type: L2
Tenant VRF: custom1
VxLAN interface: vxl11
VxLAN ifIndex: 9
Local VTEP IP: 10.125.0.1
Mcast group: 0.0.0.0
Remote VTEPs for this VNI:
  10.125.0.2 flood: HER
Number of MACs (local and remote) known for this VNI: 2
Number of ARPs (IPv4 and IPv6, local and remote) known for this VNI: 4
Advertise-gw-macip: Yes

```

You can note that `advertise-default-gw` keyword applied to VNI configuration transmits RT2 entries informing

about the IPs available on br11 interface. Also, because flooding mode is enabled by default, you can note RT3 entries.

```

rt1> show bgp l2vpn evpn
BGP table version is 5, local router ID is 10.125.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-2 prefix: [2]:[EthTag]:[MAClen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]

  Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 65500:11
*> [2]:[0]:[48]:[62:20:86:d0:e1:01]:[32]:[10.50.0.1]
      10.125.0.1          32768 i
      ET:8 RT:65500:268435467 Default Gateway
*> [2]:[0]:[48]:[62:20:86:d0:e1:01]:[128]:[fe80::7ce2:20ff:fe03:9ba]
      10.125.0.1          32768 i
      ET:8 RT:65500:268435467 Default Gateway ND:Router Flag
*>i [2]:[0]:[48]:[be:21:5b:3e:20:7b]:[32]:[10.50.0.2]
      10.125.0.2          100      0 i
      RT:65500:268435467 ET:8 Default Gateway
*>i [2]:[0]:[48]:[be:21:5b:3e:20:7b]:[128]:[fe80::bc21:5bff:fe3e:207b]
      10.125.0.2          100      0 i
      RT:65500:268435467 ET:8 Default Gateway ND:Router Flag
* i [3]:[0]:[32]:[10.125.0.1]
      10.125.0.2          100      0 i
      RT:65500:268435467 ET:8
*>
      10.125.0.1          32768 i
      ET:8 RT:65500:268435467

```

Consequently, the neighboring table is populated with local entries found locally, and remote entries learnt from BGP. For instance, 10.50.0.1 has been detected as a machine in the local network of rt1, and its MAC address and the IP association has been propagated to the remote BGP speaker.

```

rt1> show evpn arp-cache vni 11
Number of ARPs (local and remote) known for this VNI: 4
IP              Type   State   MAC              Remote VTEP      Seq #
↪ 's
10.50.0.2       remote active   be:21:5b:3e:20:7b 10.125.0.2      0/0
fe80::7ce2:20ff:fe03:9ba local  active   62:20:86:d0:e1:01
fe80::bc21:5bff:fe3e:207b remote active   be:21:5b:3e:20:7b 10.125.0.2      0/0
10.50.0.1       local  active   62:20:86:d0:e1:01

```

Also, the MAC table can be seen:

```

rt1> show evpn mac vni 11
Number of MACs (local and remote) known for this VNI: 2
MAC                Type    Intf/Remote VTEP      VLAN  Seq #'s
be:21:5b:3e:20:7b remote 10.125.0.2           0/0
62:20:86:d0:e1:01 local  br11                 1     0/0

```

It is possible to extend the layer 2 network by having a private network behind `eth0`. On the other way, VTEPs can be increased by multiplying the number of BGP peers and using route-reflector peers. While BUM traffic will be transmitted as if it was a local switch engine, if MAC table gets populated with the right MAC address, then traffic will be transmitted accordingly.

## RPKI In BGP

As BGP plays an important role in the backbone infrastructure, it is important to secure it, so as to ensure for every router the validity of learned prefixes.

There are different methods to secure exchanged BGP prefixes. The one that is presented here is a method that relies on a server/client model. The server is part of an RPKI (Resource Public Key Infrastructure), as it stands for a trusted cache server. The client is a BGP device that initiates a specific connection to the server, where Route Origin Authorizations (ROA (Route Origin Authorization)) are stored, and downloads them. Subsequently, any device can do prefix origin validation by referring to that server storage.

The specific connection done between a BGP device and the server cache uses a standard mechanism that maintains the exchange of the prefix/origin AS mapping between the cache server and routers. This is the RPKI protocol defined by **RFC 6810** (<https://tools.ietf.org/html/rfc6810.html>).

## RPKI Cache Configuration

RPKI/RTR (Resource Public Key Infrastructure to Router Protocol) cache configuration is made up of a list of servers indexed in order of preference; the lowest index representing the most preferred connection that will be tried to sync with. If the server is not reachable, then the next server with the closest preference is tested; and so on, until the connection succeeds and the resync mechanism applies.

RPKI/RTR protocol may optionally run over a secure connection. For that, an encryption protocol should be set on both ends (on the server and the client), with the appropriate keying material and authorizations.

An example of cache-server configuration for RPKI looks like the following:

```

vrouter> show config nodelist / vrf customer1 routing bgp rpki
rpki
  cache-server 1 address XXX.XXX.XXX.XXX tcp port xxxxx
  ..

```

To check if the RPKI/RTR connection did succeed, use the following command:

```
vrouters> show bgp rpki cache-connection vrf customer1
Connected to group 1
rpki tcp cache XXX.XXX.XXX.XXX xxxxx pref 1
```

It is possible to dump the prefix table by using the following command:

```
vrouters> show bgp rpki prefix-table vrf customer1
host: XXX.XXX.XXX.XXX port: xxxxx
RPKI/RTR prefix table
Prefix                               Prefix Length  Origin-AS
1.180.0.0                             14 - 14       4134
[.]
2a01:c000::                             19 - 48       5511
2003::                                 19 - 19       3320
Number of IPv4 Prefixes: 90631
Number of IPv6 Prefixes: 15573

vrouters> show bgp rpki prefix-table 91.223.245.0/24 vrf customer1
Prefix                               Prefix Length  Origin-AS
91.223.245.0                           24 - 24       48415
```

It is also possible to remove the RPKI/RTR synchronization either by removing the cache server entry, or by removing the whole RPKI configuration:

```
vrouters running config# del vrf customer1 routing bgp rpki
```

## RPKI Crypto Material

The Turbo Router offers the possibility to secure an RPKI/RTR connection using SSH. For that, the cache server must be set as an SSH server, with the appropriate authorizations and users; and the client must have a public/private key in order to authenticate to the server without interactively typing a password.

The cryptographic keys can be generated from the CLI as follows:

```
vrouters> cmd bgp rpki ssh-key create name keyfor_rpki type rsa-4096
```

Asymmetric key can either be RSA (Rivest Shamir Adleman algorithm), ECDSA (Elliptic Curve Digital Signature Algorithm) or EDDSA (Edwards Curve Digital Signature Algorithm), and are indexed by a keyword. It is possible to configure some parameters like key length for RSA (1024, 2048 or 4096 bits), or ECDSA (256, 384, or 512 bits). However, EDDSA is not customizable, and its key length is automatically set to 25519 bits.

Keys stored in the system can be listed with:

```
vrouters> cmd bgp rpki ssh-key list
keyfor_rpki
```

Public keys stored in the system can be displayed with:

```
vrouter> cmd bgp rpki ssh-key list detail
keyfor_rpki
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDM3lBagZttpcBmZUNsoaka4/WigSgZkAj5msHhqV4f/95e/
↳6GCFzCi/1EyTxRWk+eNPJfQb81Hh/
↳+edLqHpmS0zwyV9qjf2y09vbA0DkhoHhH23UBBu3KvweUpvXnfj4aQlqdsSHLE9td1Mt1NBrPXvzA1aKzzGQ1lFeNvN5z4Y
↳CSKFSl6RUHN3esPvV8I+tZT3MVYxRP91NRutESabuahyif+v2CxVngaqhjcrG1IiVK7BuZBpXTHUit7fKMfwx9XxIhhl+s+u
↳eo5EN0RN8lpN+fFCyVuI8V3r2Qds80lwVP03GHEzNc+gsKEbnp5ghM543ChfjoUtTGtbn5fW/
↳Nf0m+VLV6nWmZ7890Zx6d5qFqShLCPZ3GEenzJ/R5wZA9U+l/
↳LbtfWJW9jwPYUfMbzsm0BeAK15KiHhh0QMRnc/H0c1shD+R6zgQdMn35kdjapXSnFK2yBqb/
↳4xw5t1P8Gb0AslP9yewpq4qjTw== root@vrouter
```

And can be deleted using:

```
vrouter> cmd bgp rpki ssh-key delete keyfor_rpki
```

The public key should then be copied to the SSH server's list of authorized\_keys, under a user account created for RPKI/RTR connections.

In addition to forging local key material, it may be necessary to learn the public keys of remote servers. This can be done using the following command that populates the database with the keys of all the SSH hosts:

```
vrouter> cmd bgp rpki ssh-host add XXX.XXX.XXX.XXX port xxxxx vrf customer1
```

In case it is necessary to flush entries associated to a particular host from the local database, the following command can be used:

```
vrouter> cmd bgp rpki ssh-host delete XXX.XXX.XXX.XXX
```

Once the underlying SSH connection can be established from client to server using the newly generated keys, the secure RPKI/RTR connection can be set like below:

```
vrouter# / vrf customer1 routing bgp rpki
vrouter running rpki# cache-server 1 address XXX.XXX.XXX.XXX ssh port 22 key keyfor_
↳rpki user-name rpki-user
vrouter running rpki# commit
Configuration committed.
vrouter running rpki# show bgp rpki cache-server
host: XXX.XXX.XXX.XXX port: 22 username: rpki-user server_hostkey_path: /var/lib/yams/
↳routing/known_hosts client_privkey_path: /var/lib/yams/routing/keyfor_rpki
vrouter running rpki# show bgp rpki cache-connection
Connected to group 1
rpki ssh cache XXX.XXX.XXX.XXX 22 pref 1
```

## RPKI Miscellaneous

It is possible to configure the retry interval triggered when a server becomes unreachable. By default, retries are issued each 300 seconds. It is possible to modify this value using the following command:

```
vrouter running config# vrf customer1 routing bgp rpki retry-interval 400
```

It is also possible to configure an expiration interval for unreachable servers. The interval is set, by default, to 7200 seconds. It can be modified by issuing the following command:

```
vrouter running config# vrf customer1 routing bgp rpki expire-interval 7500
```

Similarly, it is possible to configure the polling period between two consecutive refreshes, to update the list of prefixes. Default is 3600 seconds. It can be modified by issuing the following command:

```
vrouter running config# vrf customer1 routing bgp rpki polling-period 3700
```

## RPKI Validation

Routing decisions can be made based on RPKI route announcement validity. They are implemented at BGP neighbor level, by the means of a route-map, applied to the ingress direction. The keywords `rpki valid`, `rpki invalid` or `rpki notfound` can be invoked in a given route-map sequence, thus triggering a policy when matched.

The below example depicts what can be done in order to accept valid prefixes; reject invalid prefixes; and set a low local preference to unknown prefixes, as they can't be fully trusted.

```
routing
  route-map rmap
    seq 1
      policy permit
      match rpki valid
      ..
    seq 2
      policy deny
      match rpki invalid
      ..
    seq 3
      policy permit
      match rpki notfound
      set local-preference 20
      ..
  / vrf customer1
    routing bgp
      as 1
      neighbor 10.1.1.11 address-family ipv4-unicast route-map in route-map-name rmap
```

## MPLS

### LDP

#### LDP Overview

The LDP daemon is a standardised protocol that permits exchanging MPLS label information between MPLS capable devices (also called LSRs (Label-Switched Routers) or LERs (Label Edge Routers)). The LDP protocol creates peering between devices, so as to exchange FEC (Forwarding Equivalence Class) information linking networks and labels together. This information is stored in MPLS binding tables. Then, based on available routing table, on the one hand, an MPLS label is appended to the routing table for locally generated packets; on the other hand, LFIB entry is created with that label information for incoming MPLS frames. By acting this way, data traffic is encapsulated in MPLS header, and on each LSR, the incoming label will determine which label has to be swapped, instead of the former one. MPLS permits carrying any transportation data through that MPLS backbone. It is possible to carry L3VPN traffic inside, thus permitting transporting overlapping data to different place. LDP often works in conjunction with IGP routing protocols like OSPF, thus facilitating the discovery of the backbone, and permitting to know for some traffic having to go through that backbone what is the best path to take.

The LDP is handled by FRR (<https://frouting.org/>).

#### LDP packets and operations

LDP aims at sharing label information across devices. It tries to establish peering with remote LDP capable devices, first by discovering using UDP port 646 and multicast address 224.0.0.2 on the connected nodes, then by peering using TCP port 646. Once the TCP session is established, the following happens:

---

**Note:** This is the standard way for LDP sessions to peer together in a LSR. For that, LDP peers are directly connected, so that a label can be assigned to each IGP route. Let us remind that label switching of packets is hop per hop.

---

- The list of IP addresses that each device owns is sent via an **Address Message** LDP message. Knowing local IP addresses permits to know that this address is reachable via an **implicit-null** message.
- Then **Label Mapping Messages** are sent. Those messages stand for each known network associatet to a label. This is the so-called FEC information, where usually a destination IP prefix is bound to a label. It is worth to be noted that when a route is connected, the label associated is **implicit-null**, which is 3 in the case of IPv4. For the other routes, an arbitrary label value is communicated. A **binding table** is then forged on each device, and contains, for each route, the local label emitted for each network, and the remote label received from remote peer. This table is then used to apply labels to already present routes.

There are different methods to send label advertisement modes. The implementation actually supports the following : **Liberal Label Retention + Downstream Unsolicited + Independent Control**. The other advertising modes are depicted below, and compared with the current implementation.

- Liberal label retention versus conservative mode: In liberal mode, every label sent by every LSR is stored in the MPLS table. In conservative mode, only the label that was sent by the best next hop (determined by the IGP metric) for that particular FEC is stored in the MPLS table.
- Independent LSP Control versus ordered LSP Control MPLS has two ways of binding labels to FEC; either through ordered LSP control, or independent LSP control. Ordered LSP control only binds a label to a FEC if it is the egress LSR, or the router received a label binding for a FEC from the next hop router. In this mode, an MPLS router will create a label binding for each FEC and distribute it to its neighbors so long as he has an entry in the RIB for the destination. In the other mode, label bindings are made without any dependencies on another router advertising a label for a particular FEC. Each router makes its own independent decision to create a label for each FEC. By default IOS uses Independent LSP Control, while Juniper implements the Ordered Control. Both modes are interoperable, the difference is that Ordered Control prevents blackholing during the LDP convergence process, at the cost of slowing down the convergence itself.
- unsolicited downstream versus downstream on demand Downstream on demand label distribution is where an LSR must explicitly request that a label be sent from its downstream router for a particular FEC. Unsolicited label distribution is where a label is sent from the downstream router without the original router requesting it.

LDP has by default the PHP (Penultimate Hop Popping) functionality. That functionality stipulates that the outermost label of an MPLS tagged packet is removed by a LSR before the packet is passed to an adjacent LER. This behaviour permits sparing one CPU cycle on the LER, by not popping that last label. However, LDP provides the configuration means to disable that feature, by using explicit-null labels.

## RFC

**RFC 5036** (<https://tools.ietf.org/html/rfc5036.html>): LDP specification

**RFC 5082** (<https://tools.ietf.org/html/rfc5082.html>): The Generalized TTL Security Mechanism (GTSM)

**RFC 6720** (<https://tools.ietf.org/html/rfc6720.html>): The Generalized TTL Security Mechanism for the LDP

**RFC 6667** (<https://tools.ietf.org/html/rfc6667.html>): LDP ‘Typed Wildcard’ Forwarding Equivalence Class (FEC) for PwId and Generalized PwId FEC Elements

**RFC 5919** (<https://tools.ietf.org/html/rfc5919.html>): Signaling LDP Label Advertisement Completion

**RFC 5561** (<https://tools.ietf.org/html/rfc5561.html>): LDP capabilities

**RFC 7552** (<https://tools.ietf.org/html/rfc7552.html>): Updates to LDP for IPv6

### See also:

The *command reference* for details.

## LDP configuration

There are a list of necessary elements to know when forging a LDP configuration.

- *Basic elements for configuration*
- *Basic LDP configuration*
- *LDP Disabling PHP*
- *LDP Interoperability*
- *BackBone LDP configuration*

### Basic elements for configuration

When forging a LDP configuration, a router-id has to be defined. It is usually the IP address of one loopback interface. Then the address-family where LDP will operate the discovery has to be configured, as well as the interfaces, and the IP transportation to use.

Here below is an example on how to configure a sample LDP configuration with IPv4 address-family set:

```
vrf main
  routing mpls ldp
    router-id 5.5.5.5
    address-family ipv4
      discovery transport-address 5.5.5.5
      interface eth0_0
      ..
    ..
  ..
..
commit
```

---

**Note:** You can also disable LDP, either by suppressing the configuration:

---

```
vrf main
  del routing mpls ldp
  ..
```

Alternatively, if you don't want to lose the configuration, and disabling LDP configuration, you can use following command:

```
vrf main
  routing mpls ldp
    enabled false
```

This method can be used if the user wants to force the reset of LDP configuration.

```
vrf main
  routing mpls ldp enabled false
  commit
  routing mpls ldp enabled true
  commit
```

### Basic LDP configuration

Instantiating a basic back to back configuration setup between two devices is a first step towards understanding LDP but is not enough. Below configuration illustrates this, with `rt1` and `rt2` configurations. The basic neighbour discovery mechanism is used to make the peering work. You can note that LDP operates over loopback interfaces, like most IGP do. Following drawing illustrates the networking topology and information.

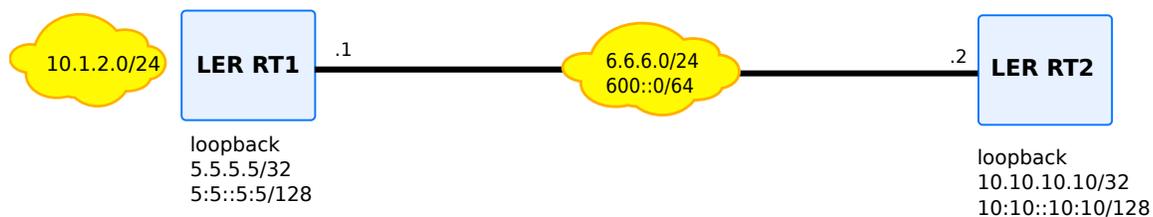


Fig. 10: LDP back to back topology between 2 LER

**rt1**

```
vrf main
  routing mpls ldp
    router-id 5.5.5.5
    dual-stack transport-preference ipv4
    address-family ipv4
      discovery transport-address 5.5.5.5
      interface eth0_0
      ..
      ..
      ..
    address-family ipv6
      discovery transport-address 5:5::5:5
      interface eth0_0
      ..
      ..
      ..
      ..
      ..
  routing static
    ipv4-route 10.10.10.10/32 next-hop 6.6.6.2
    ipv6-route 10:10::10:10/128 next-hop 6000::2
    ..
    ..
  interface
    loopback loop1
      ipv4 address 5.5.5.5/32
      ipv6 address 5:5::5:5/128
      ..
    physical eth0_0
      ipv4 address 6.6.6.1/24
      ipv6 address 600::1/64
      ..
      ..
```

**rt2**

```
vrf main
  routing mpls ldp
    router-id 10.10.10.10
    dual-stack transport-preference ipv4
    discovery hello holdtime 2
    discovery hello interval 2
    address-family ipv4
      discovery transport-address 10.10.10.10
      interface eth0_0
      ..
    ..
    ..
  address-family ipv6
    discovery transport-address 10:10::10:10
    interface eth0_0
    ..
    ..
    ..
  ..
  ..
  ..
  routing static
    ipv4-route 5.5.5.5/32 next-hop 6.6.6.1
    ipv6-route 5:5::5:5/128 next-hop 6000::1
    ..
  ..
  interface
    loopback loop1
      ipv4 address 10.10.10.10/32
      ipv6 address 10:10::10:10/128
      ..
    ..
  physical eth0_0
    ipv4 address 6.6.6.2/24
    ipv6 address 6000::2/64
    ..
  ..
  ..
```

After having executed the two configurations, the status of the LDP discovery can be obtained, by using following command:

```
rt3> show mpls-ldp discovery detail
```

(continues on next page)

(continued from previous page)

```

Local:
  LSR Id: 5.5.5.5:0
  Transport Address (IPv4): 5.5.5.5
  Transport Address (IPv6): 5:5::5:5
Discovery Sources:
  Interfaces:
    r1-eth2:
      LSR Id: 10.10.10.10:0
      Source address: 6.6.6.2
      Transport address: 10.10.10.10
      Hello hold time: 15 secs (due in 14 secs)
      Dual-stack capability TLV: yes
      LSR Id: 10.10.10.10:0
      Source address: fe80::d0fc:e8ff:fee0:86dd
      Transport address: 10:10::10:10
      Hello hold time: 15 secs (due in 11 secs)
      Dual-stack capability TLV: yes
  Targeted Hellos:

```

Also, to know about the status of the peering connections, there is a specific command for that (see below). You can note that the two neighbors successfully peered together, as you can see that the state of the connection is OPERATIONAL. The discovery process on UDP port 646 resulted in creating a TCP session between both sides. Subsequently, destination prefixes and labels were exchanged.

```

rt1> show mpls-ldp neighbor
AF  ID                State      Remote Address  Uptime
ipv4 10.10.10.10        OPERATIONAL 10.10.10.10    00:01:32

```

Also, it is possible to visualise the configured interfaces.

```

rt1> show mpls ldp interface
AF  Interface  State  Uptime  Hello Timers  ac
ipv4 eth0_0    ACTIVE 00:15:44 4/15        0
ipv6 eth0_0    ACTIVE 00:15:43 4/15        0

```

It is worth to be noted that the destination prefixes exchanges rely on the address family to be configured. Not configuring it will result in not having destination prefixes of that address-family. Also, if chosen, the discovery transport-address is necessary. Also, it is worth to be noted that LDP protocol plans to use ipv6 if both address-families are chosen. To mitigate this, an extra command has been added (`dual-stack transport-preference ipv4`) to the configuration so as to fallback over ipv4.

The above configuration results in having the following list of bindings. As said previously, the establishment of TCP sessions between ldp peers, leads to exchange of label mapping messages that permit exchange FEC. In our usage, the FEC stands for the relationship between a Label and a destination IP. The Local Label column stands for locally generated messages and outgoing label to be applied, while Remote Labels stand for received

messages from peers. The remote label information is used to establish switching rules. Nexthop stands for the ldp remote peer IP address that sent the message.

```

rt1> show mpls-ldp binding
AF  Destination          Nexthop          Local Label Remote Label  In Use
ipv4 5.5.5.5/32           10.10.10.10     imp-null      16             no
ipv4 10.10.10.10/32      10.10.10.10     16           imp-null      yes
ipv6 5:5::5:5/128       10.10.10.10     imp-null      17             no
ipv6 10:10::10:10/128   10.10.10.10     17           imp-null      yes

rt1> show mpls-ldp binding ipv6
AF  Destination          Nexthop          Local Label Remote Label  In Use
ipv6 5:5::5:5/128       10.10.10.10     imp-null      17             no
ipv6 10:10::10:10/128   10.10.10.10     17           imp-null      yes

```

The In Use column tells if the system has routing or switching support to add label information or not. 5.5.5.5/32 and 5:5::5:5/128 are local networks, so do not need to be used with MPLS. However, as we have static routes for reaching 10.10.10.10/32 and 10:10::10:10/128 networks, the binding MPLS information will be used. Practically, an implicit-null label is appended to route entry.

```

rt1> show ipv4-routes
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route
[...]
S>* 10.10.10.10/32 [1/0] via 6.6.6.2, eth0_0, label implicit-null, 00:08:17

rt1> show ipv6-routes
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, N - NHRP, T - Table,
       v - VNC, V - VNC-Direct, A - Babel, D - SHARP, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route
[...]
S>* 10:10::10:10/128 [1/0] via 6000::2, eth0_0, label implicit-null, 00:08:19

```

Also, an MPLS switching entry is put in place, so that incoming MPLS packets coming with a label of 16 or 17 be switched with an implicit-null label to go to respective nexthop 6.6.6.2 and 6000::2.

```
rt1> show mpls table
```

```

Inbound Outbound Label Type Nexthop Label

```

```

-----
-----
-----
-----
----- 16 LDP 6.6.6.2 implicit-null 17 LDP 6000::2 implicit-null

```

## LDP Disabling PHP

Above example illustrates that back to back configurations, the label used is an `implicit-null` label. That label is used when the nexthop is adjacent, that is to say connected. This is called the penultimate hop popping feature. PHP feature avoids having an outermost label between the last LSR and the LER where traffic is heading to. However, that feature can be interesting to disable on some cases. For instance, when working on a back to back operating mode. Below example gives an example on how explicit-null labels can be configured instead of using implicit-null labels on the LER side.

**rt1**

```
vrf main
  routing mpls ldp
    router-id 5.5.5.5
    address-family ipv4
      discovery transport-address 5.5.5.5
      label local advertise explicit-null
    interface eth0_0
      ..
    ..
    ..
  ..
  ..
  ..
  routing static
    ipv4-route 10.10.10.10/32 next-hop 6.6.6.2
    ..
  ..
  interface
    loopback loop1
      ipv4 address 5.5.5.5/32
      ..
    physical eth0_0
      ipv4 address 6.6.6.1/24
      ..
    ..
```

**rt2**

```
vrf main
[.]
routing mpls ldp
  router-id 10.10.10.10
  dual-stack transport-preference ipv4
  address-family ipv4
    discovery transport-address 10.10.10.10
    label local advertise explicit-null
  [.]
```

On the peer router receiving the LDP advertisements, an `explicit-null` label is received, associated with the `10.10.10.10` next-hop address.

```
rt2> show mpls-ldp binding
AF  Destination      Nexthop      Local Label Remote Label  In Use
ipv4 5.5.5.5/32        5.5.5.5      16           exp-null      yes
ipv4 5.5.5.5/32        5.5.5.5      exp-null     16           no

rt2> show mpls table
Inbound                                Outbound
Label  Type      Nexthop      Label
-----
  16    LDP      10.125.0.2  IPv4 Explicit Null
```

**Note:** `explicit-null` label must be only used if it is the last label, that is to say that the label will have BOS bit. In other case will trigger packet drops ( as per [RFC 3032](https://tools.ietf.org/html/rfc3032) (<https://tools.ietf.org/html/rfc3032>)). Example scenario where that value can be used will only involve LDP, not L3VPN with multiple stacking.

**LDP Interoperability**

LDP specification stipulates to use ipv6 transporation when both address-families are negotiated. Adding to this, Cisco uses a non-compliant format to send and interpret the dual-stack capabilities TLV contained in LDP packets. For that, it is possible to align with cisco behaviour and a configuration command is available :

```
vrf main
  routing mpls ldp
    router-id 10.10.10.10
    dual-stack cisco-interop true
  address-family ipv4
    discovery transport-address 10.10.10.10
```

(continues on next page)

(continued from previous page)

```

interface eth0_0
  ..
  ..
  ..
address-family ipv6
  discovery transport-address 10:10::10:10
  interface eth0_0
    ..
    ..
    ..
    ..
  
```

### BackBone LDP configuration

LDP Backbone Illustration

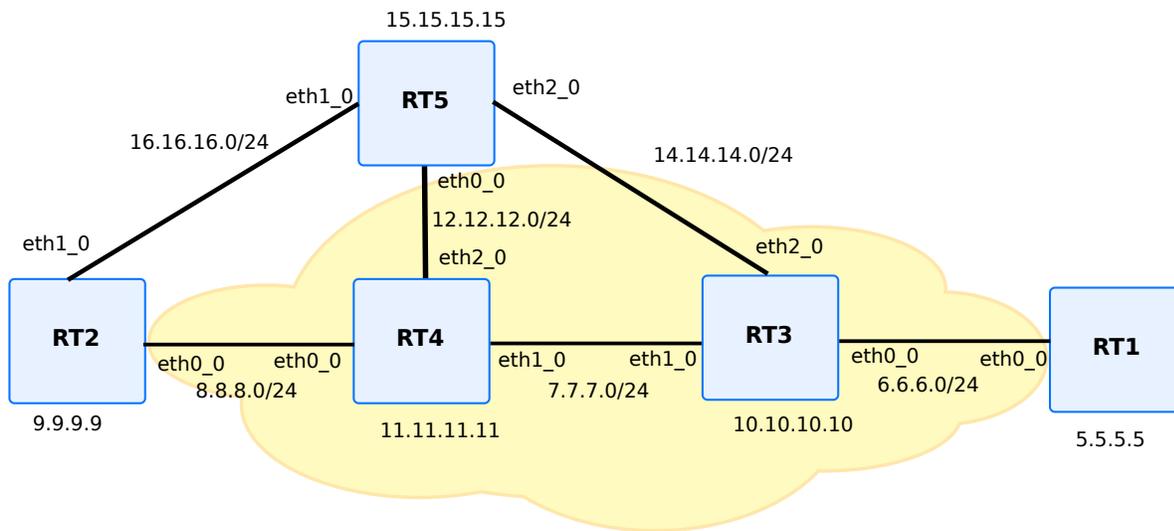


Fig. 11: LDP backbone illustration with multiple nodes, and multiple paths

Following setup illustrates what a backbone looks like. Actually, to prevent from link failure or node failure, you can see that there are several paths available to link some nodes together. For instance, to link `rt1` with `rt2`, either `rt5` or `rt4` can be used, thus preventing from link failure. Also, to prevent from `r4` node failure, you can note that there is a path that links `rt2` to `rt3` by relying on `rt5` instead.

By default, multipath is enabled. That implies that unless you rely on some IGP like OSPF to help in finding out some routing decisions, available paths will be equal. ( for example, lowering the bandwidth or configuring the cost of the interface between `rt2` and `rt5` will trigger in proposing only one route).

The above diagram relies on both OSPF and LDP routing daemons. OSPF is used for IP discovery, while LDP will allocate labels for LSR and LER. Below is shown the aggregated LDP and OSPF configuration.

### rt1

```
routing ospf
  router-id 5.5.5.5
  network 5.5.5.5/32 area 0
  network 6.6.6.0/24 area 0
  passive-interface loop1
  ..
  ..
routing mpls ldp
  router-id 5.5.5.5
  address-family ipv4
    discovery transport-address 5.5.5.5
    interface eth0_0
      ..
    ..
  ..
  ..
  ..
  ..
  ..
interface
  loopback loop1
    ipv4 address 5.5.5.5/32
    ..
  ..
  physical eth0_0
    ipv4 address 6.6.6.1/24
    ..
  ..
```

### rt2

```
routing ospf
  router-id 9.9.9.9
  network 9.9.9.9/32 area 0
  network 8.8.8.0/24 area 0
  network 16.16.16.0/24 area 0
  passive-interface loop1
  ..
  interface eth1_0
```

(continues on next page)

(continued from previous page)

```
    ip ospf cost 100
    ..
    ..
routing mpls ldp
router-id 9.9.9.9
address-family ipv4
  discovery transport-address 9.9.9.9
  interface eth0_0
    ..
  interface eth1_0
    ..
    ..
    ..
    ..
    ..
interface
  loopback loop1
    ipv4 address 9.9.9.9/32
    ..
    ..
  physical eth0_0
    ipv4 address 8.8.8.2/24
    ..
    ..
  physical eth1_0
    ipv4 address 16.16.16.2/24
    ..
    ..
```

**rt3**

```
routing ospf
  router-id 10.10.10.10
  network 10.10.10.10/32 area 0
  network 6.6.6.0/24 area 0
  network 7.7.7.0/24 area 0
  passive-interface loop1
  ..
  ..
routing mpls ldp
  router-id 10.10.10.10
```

(continues on next page)

(continued from previous page)

```
address-family ipv4
  discovery transport-address 10.10.10.10
  interface eth0_0
    ..
  interface eth1_0
    ..
  interface eth2_0
    ..
  ..
  ..
  ..
  ..
  ..
  ..
interface
  loopback loop1
    ipv4 address 10.10.10.10/32
    ..
  ..
  physical eth0_0
    ipv4 address 6.6.6.3/24
    ..
  ..
  physical eth1_0
    ipv4 address 7.7.7.3/24
    ..
  ..
  physical eth2_0
    ipv4 address 14.14.14.3/24
    ..
  ..
```

#### rt4

```
routing ospf
  router-id 11.11.11.11
  network 11.11.11.11/32 area 0
  network 12.12.12.0/24 area 0
  network 7.7.7.0/24 area 0
  passive-interface loop1
  ..
  ..
routing mpls ldp
```

(continues on next page)

(continued from previous page)

```
router-id 11.11.11.11
address-family ipv4
  discovery transport-address 11.11.11.11
  interface eth0_0
    ..
  interface eth1_0
    ..
  interface eth2_0
    ..
  ..
  ..
  ..
  ..
  ..
  ..
interface
  loopback loop1
    ipv4 address 11.11.11.11/32
    ..
  ..
  physical eth0_0
    ipv4 address 8.8.8.4/24
    ..
  ..
  physical eth1_0
    ipv4 address 7.7.7.4/24
    ..
  ..
  physical eth2_0
    ipv4 address 12.12.12.4/24
    ..
  ..
```

**rt5**

```
routing ospf
  router-id 15.15.15.15
  network 15.15.15.15/32 area 0
  network 12.12.12.0/24 area 0
  network 16.16.16.0/24 area 0
  network 14.14.14.0/24 area 0
  passive-interface loop1
  ..
```

(continues on next page)

(continued from previous page)

```

interface eth1_0
  ip ospf cost 100
  ..
  ..
routing mpls ldp
router-id 15.15.15.15
address-family ipv4
  discovery transport-address 15.15.15.15
  interface eth0_0
    ..
  interface eth1_0
    ..
  interface eth2_0
    ..
  ..
  ..
  ..
  ..
  ..
interface
  loopback loop1
    ipv4 address 15.15.15.15/32
    ..
  ..
  physical eth0_0
    ipv4 address 12.12.12.5/24
    ..
  ..
  physical eth1_0
    ipv4 address 16.16.16.5/24
    ..
  ..
  physical eth2_0
    ipv4 address 14.14.14.5/24
    ..
  ..

```

After having executed the above configurations, the status of the LDP connections can be obtained. The peerings between the devices can be visualised with the following command:

```

rt3> show mpls-ldp neighbor
AF  ID           State           Remote Address   Uptime
ipv4 9.9.9.9       OPERATIONAL    9.9.9.9         00:13:15
ipv4 10.10.10.10   OPERATIONAL    10.10.10.10    00:13:15

```

(continues on next page)

(continued from previous page)

|                  |             |             |          |
|------------------|-------------|-------------|----------|
| ipv4 15.15.15.15 | OPERATIONAL | 15.15.15.15 | 00:13:05 |
|------------------|-------------|-------------|----------|

It is possible to get the whole list of bindings that LDP made, on each IP route. As LDP obtains labels for all networks, those labels are bound and installed, upon availability of associated network entries on the underlying system. The redistributed OSPF routes are then useful for that.

```
rt2> show mpls-ldp binding
```

| AF   | Destination    | Nextthop    | Local Label | Remote Label | In Use |
|------|----------------|-------------|-------------|--------------|--------|
| ipv4 | 5.5.5.5/32     | 11.11.11.11 | 22          | 21           | yes    |
| ipv4 | 6.6.6.0/24     | 11.11.11.11 | 19          | 18           | yes    |
| ipv4 | 7.7.7.0/24     | 11.11.11.11 | 16          | imp-null     | yes    |
| ipv4 | 8.8.8.0/24     | 11.11.11.11 | imp-null    | imp-null     | no     |
| ipv4 | 9.9.9.9/32     | 11.11.11.11 | imp-null    | 16           | no     |
| ipv4 | 10.10.10.10/32 | 11.11.11.11 | 20          | 19           | yes    |
| ipv4 | 11.11.11.11/32 | 11.11.11.11 | 17          | imp-null     | yes    |
| ipv4 | 12.12.12.0/24  | 11.11.11.11 | 18          | imp-null     | yes    |
| ipv4 | 14.14.14.0/24  | 11.11.11.11 | 21          | 20           | yes    |
| ipv4 | 15.15.15.15/32 | 11.11.11.11 | 23          | 22           | yes    |
| ipv4 | 16.16.16.0/24  | 11.11.11.11 | imp-null    | 17           | no     |

Note that some entries are not in use, since OSPF did choose to prefer rt4 link over rt5 link. Subsequently, it is also possible what are the bindings currently installed on the system:

```
rt2> show ipv4-routes vrf main
```

```
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

O>* 5.5.5.5/32 [110/30] via 8.8.8.4, r2-eth2, label 21, 00:22:09
O>* 6.6.6.0/24 [110/30] via 8.8.8.4, r2-eth2, label 18, 00:22:16
O>* 7.7.7.0/24 [110/20] via 8.8.8.4, r2-eth2, label implicit-null, 00:22:16
O 8.8.8.0/24 [110/10] is directly connected, r2-eth2, 00:22:17
C>* 8.8.8.0/24 is directly connected, r2-eth2, 00:23:02
O 9.9.9.9/32 [110/0] is directly connected, lo, 00:23:01
C>* 9.9.9.9/32 is directly connected, lo, 00:23:02
O>* 10.10.10.10/32 [110/20] via 8.8.8.4, r2-eth2, label 19, 00:22:16
O>* 11.11.11.11/32 [110/10] via 8.8.8.4, r2-eth2, label implicit-null, 00:22:16
O>* 12.12.12.0/24 [110/20] via 8.8.8.4, r2-eth2, label implicit-null, 00:22:16
O>* 14.14.14.0/24 [110/30] via 8.8.8.4, r2-eth2, label 20, 00:22:16
O>* 15.15.15.15/32 [110/20] via 8.8.8.4, r2-eth2, label 22, 00:22:06
O 16.16.16.0/24 [110/100] is directly connected, r2-eth3, 00:23:01
C>* 16.16.16.0/24 is directly connected, r2-eth3, 00:23:02
```

It is also possible to dump the contexts of the LSR. For instance, on `rt3` or `rt4`, one can see the LFIB:

```
rt4> show mpls table
```

| Inbound<br>Label | Type | Nexthop     | Outbound<br>Label |
|------------------|------|-------------|-------------------|
| 16               | LDP  | 8.8.8.2     | implicit-null     |
| 17               | LDP  | 12.12.12.12 | implicit-null     |
| 17               | LDP  | 8.8.8.2     | implicit-null     |
| 18               | LDP  | 7.7.7.3     | implicit-null     |
| 19               | LDP  | 7.7.7.3     | implicit-null     |
| 20               | LDP  | 12.12.12.12 | implicit-null     |
| 20               | LDP  | 7.7.7.3     | implicit-null     |
| 21               | LDP  | 7.7.7.3     | 21                |
| 22               | LDP  | 12.12.12.12 | implicit-null     |

## LDP security

LDP is a critical service for the internet infrastructure. Security aspects for LDP are important.

### LDP Neighbor Security

In order to avoid peering with unexpected neighbors, it is possible to configure a password on both sides. A TCP MD5 digest is then calculated, thus preventing to create a peering with a misconfigured peer.

```
vrf main
  routing mpls ldp
    router-id 10.10.10.10
    neighbor 5.5.5.5 password secret_phrase
    address-family ipv4
      discovery transport-address 10.10.10.10
    interface eth0_0
      ..
    ..
  ..
..
```

## LDP TTL security

**RFC 6720** (<https://tools.ietf.org/html/rfc6720.html>) stipulates that only nodes from connected links are considered as accepted, when it comes to LDP peering with basic discovery mode. This is where ttl-security acts, since it ensures that the node is really connected, by not only looking up the ttl value, but also appending some values on the LDP options. It is however possible to disable that security check in some cases, for instance, to keep compatibility with old **RFC 5082** (<https://tools.ietf.org/html/rfc5082.html>). To disable ttl-security checking, use the following command:

```
vrf main
  mpls ldp
    router-id 10.10.10.10
    neighbor 5.5.5.5 ttl-security disable true
    address-family ipv4
      discovery transport-address 10.10.10.10
      interface eth0_0
      ..
    ..
  ..
..
..
..
```

## LDP filtering

There are some set of commands that permit filtering the LDP behavior, either by filtering incoming requests or filtering outgoing requests. For instance, it is possible to accept incoming ipv4 or ipv6 incoming, by filtering based on the remote LDP peer. Below configuration illustrates this:

```
vrf main
  routing mpls ldp
    router-id 10.10.10.10
    address-family ipv4
      discovery transport-address 10.10.10.10
      interface eth0_0
      ..
      label remote accept from 11
      ..
    ..
  ..
..
..
..
```

(continues on next page)

(continued from previous page)

```

routing
  ipv4-access-list 11 permit 10.10.10.10/32

```

It is also possible to apply filtering on incoming requests, based on the incoming destination prefixes, like suggests below configuration with an incoming prefix 4.4.4.0/24.

```

vrf main
  routing mpls ldp
    router-id 10.10.10.10
    address-family ipv4
      discovery transport-address 10.10.10.10
      interface eth0_0
      ..
      label remote accept for 12
      ..
      ..
    ..
    ..
    ..
  ..
routing
  ipv4-access-list 12 permit 4.4.4.0/24

```

It is also possible to apply filtering on the allocated labels. Locally, a label may be allocated only for host routes, thus sparing labels.

```

vrf main
  routing mpls ldp
    router-id 10.10.10.10
    address-family ipv4
      discovery transport-address 10.10.10.10
      interface eth0_0
      ..
      label local allocate host-routes
      ..
      ..
    ..
    ..
    ..
  ..

```

Adding to this, if it is not enough, it is also possible to control the allocation of labels by explicitly listing the destination prefixes that should gain a binding.

```
vrf main
  routing mpls ldp
    router-id 10.10.10.10
    address-family ipv4
      discovery transport-address 10.10.10.10
      interface eth1_0
      ..
      interface eth0_0
      ..
      label local allocate for 13
      ..
      ..
  ..
  ..
  ..
  ..
routing
  ipv4-access-list 13 permit 2.2.2.0/24
```

Finally, it is possible to do outgoing filtering, by selecting which peer or which destination prefix deserves to be sent or not. Like below example suggests, only the destination prefix 4.4.4.0/24 will only be sent to peer 5.5.5.5.

```
vrf main
  routing mpls ldp
    router-id 10.10.10.10
    address-family ipv4
      discovery transport-address 10.10.10.10
      interface eth0_0
      ..
      label local advertise to 14 for 15
      ..
      ..
  ..
  ..
  ..
  ..
routing
  ipv4-access-list 14 permit 5.5.5.5/32
  ipv4-access-list 15 permit 4.4.4.0/24
```

## MPLS

MPLS aims at combining the switching technique at network layer 2 of labels, with the layer 3 protocols. Nowadays, many backbone networks use MPLS as the switching technology carrying any kind of traffic. MPLS permits performance, thanks to the switching technique very close to what ATM or Frame-Relay was doing a few years ago. Initially, IP networks were carried by MPLS. Today, because any transport over MPLS is possible (ATOM (Any Transport Over MPLS)), it is also used to carry L3VPN and L2VPN traffic.

This chapter aims at explaining how MPLS works, explains the main concepts, and explains the differences with classical routing.

### MPLS terminology

It is important to understand the MPLS terminology. In this paragraph we will give the most important concepts.

**LSR** Labeled Switch Router. Networking devices handling labels used to forward traffic between and through them.

**LER** Labeled Edge Router. A Labeled edge router is located at the edge of an MPLS network, generally between an IP network and an MPLS network.

**LFIB** Label Forwarding Information Base. A data structure in which incoming interface and incoming labels are associated with outgoing interfaces and labels.

**label binding** An association between a label and a set of packets, which can be advertised to neighbors so that a label switched path can be established.

**FEC** Forwarding Equivalent Class. It is a term used in Multiprotocol Label Switching (MPLS) to describe a set of packets with similar or identical characteristics which may be forwarded the same way; that is, they may be bound to the same MPLS label. In classical IP routing, the FEC choice is usually done according to destination IP address.

**MPLS label** The MPLS label is a 4 byte field that contains a 20 bit label value, a 3 bit cos value, an 8 bit ttl value, and 1 BOS bit indicating that the label is the last one of the stack. Actually, MPLS can be stacked (then we could use the term LSP Tunneling or Label Stacking). This BOS information indicates that next payload is not an MPLS packet.

### MPLS operations

Here are the operations that are applied coming from A and going to B, through an MPLS network.

Packet will first be sent to a LER that stands for the ingress node.

On classical IP routing using Ethernet as medium, an incoming IP packet will be routed, by using its destination IP address; the FIB is inspected, a nexthop IP is returned if everything went well; then the MAC information is appended to the packet; source mac address is the mac address of the outgoing interface, while destination mac address will be obtained by using the destination mac address of the resolved nexthop.

On a LER, if the nexthop information is reachable through a MPLS network, an extra information called FEC will be located in the FIB. A Label will be *pushed* between the IP layer and the MAC layer. This extra relationship is called **label binding**.

Then, the encapsulated MPLS packet will be sent to the destination mac address indicated by its packet. It is received by an incoming LSR. Here, the LFIB is looked up, based on the incoming MPLS label. LFIB returns a *swap* operation: the incoming label will be replaced by an outgoing label; the new MPLS packet is being sent to the next hop. Before reaching the final destination, the MPLS label must be *popped*. This happens if the LFIB indicates to pop the label; for instance, the label is being replaced by an implicit-Null label. Here, the IP packet has reached the egress node.

The whole path between the ingress and the egress node is called the LSP. The incoming label set at the ingress node, will determine the whole path the packet uses to reach the egress node. By setting the appropriate FEC information at the LER, it is possible to apply specific path, depending on the characteristics of the incoming traffic. Note also that because that FEC information can be applied to all kind of traffic, one can have multiple criteria.

## Label Distribution

Establishing a LSP requires coordination between all LER and LSR. This is done by distributing protocols. For instance, this can be done by using LDP protocol. Please see *Label Distribution Protocol* for details.

## Label Stacking

Several services can rely on MPLS framework, and not only IP. One example is L3VPN technology. BGP provides the capability to exchange VPN information, by exchanging labels. Label stacking is then used. More information can be found in *BGP L3VPN*.

## RFC

**RFC 3032 (<https://tools.ietf.org/html/rfc3032.html>):** MPLS Label Stack Encoding

## DMVPN and NHRP

### DMVPN and NHRP Overview

DMVPN is a dynamic tunneling form of a VPN, that enables to create a dynamic-mesh VPN network without having to statically pre-configure all possible tunnel endpoint peers.

The DMVPN solution enables to dynamically build an NBMA (Non-Broadcast Multiple Access) network that interconnects scattered sites via GRE tunnels.

Static or dynamic GRE tunnels are used to interconnect the sites, the NHRP protocol is used to determine a route with the fewest hops from a sender to a receiver.

The DMVPN solution relies on a hub and spoke model, but supports optimizing routes, so that spokes can directly communicate together via dynamic GRE tunnels.

Finally, DMVPN supports an automatic IPSEC protection with limited configuration.

## DMVPN and NHRP terminology

**Hub and Spoke model** A computer network topology in which a series of spokes connect to a central hub. Communication between spokes transits via the hub. The hub dispatches traffic or information among the spokes.

**NBMA, Non-Broadcast Multiple-Access network** A computer network to which multiple hosts are attached, but data is transmitted only directly from one computer to another single host, typically over a virtual circuit. In the DMVPN case, the virtual circuits are GRE tunnels.

**NHRP, Next Hop Resolution Protocol** A client-server protocol used to determine a route with the fewest hops from a sender to a receiver in an NBMA network.

**NHC (Next Hop Clients), NHRP client** A client of the NHRP protocol, that runs on a spoke.

**NHS (Next Hop Servers), NHRP server** The server of the NHRP protocol, that runs on the hub.

**DMVPN, Dynamic Multipoint Virtual Private Network** A dynamic tunneling form of a VPN, that enables to create a dynamic-mesh VPN network without having to statically pre-configure all possible tunnel endpoint peers.

## DMVPN and NHRP operation

### Hub and spoke topology

A DMVPN network is an NBMA network composed of spokes (NHC devices) connected to a central hub (NHS device) via GRE tunnels.

Each device has an internal IP address, named the *protocol address*, typically configured on its GRE interface, and an outer IP address, named the *NBMA address*, which is the source address of the GRE packets.

The hub and all spokes are considered IP neighbors in the NBMA network, whose IP addresses are their protocol address.

All members of the DMVPN network register their protocol address and NBMA address to the NHS via the NHRP protocol.

The NHRP protocol enables to resolve the next-hop to reach the protocol address of a neighbor in the NBMA network, i.e. the destination IP address of GRE packets in which the packets should be encapsulated.

Depending on the situation, the next hop may be the NBMA address of the NHS (the traffic will transit via the hub) or the NBMA address of the destination spoke (if a shortcut optimization is established).

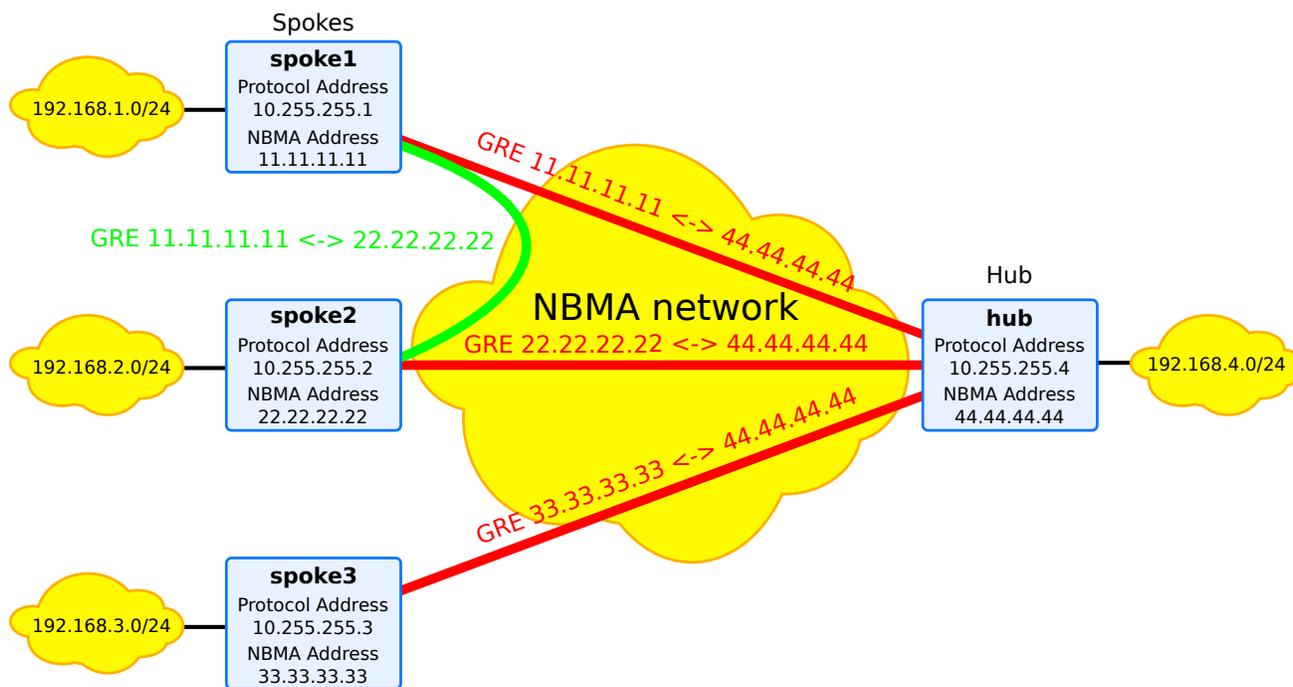


Fig. 12: NHRP use case example

The diagram above illustrates the use case that will be used across all the phases proposed. It is a NBMA network with 4 devices. One of those devices acts as the hub, while the others stand for the spokes. Here are some details:

- NBMA network spans over Internet and the nodes use public IPv4 addresses.
- each device has a GRE interface with an IP address on GRE interface named protocol address. That IP address will need to be a 32 bit mask or 128 bit mask, whether it is an IPv4 or an IPv6 address. Configuring an other bitmask will lead to a misconfiguration problem. Once done, NHRP will use that IP address to associate with its NBMA address. NHRP is able to mount dynamically or statically routes to a remote NBMA address.
- each device is attached to its own private network 192.168.y.z/24. More than one private network can be used behind GRE interface.
- a gre tunnel is established. This tunnel relies on NBMA source ip address of the device.
- Optionally, all traffic going through the gre interface is encrypted.

NHRP can help in establishing traffic between two devices, either between spoke and hub, or between two spokes. In the former case, it is possible to avoid configuring multipoint GRE interfaces, as traffic is redirected to hub.

---

**Note:** DMVPN is partially supported on cross-vrf GRE interfaces, i.e. GRE interfaces with a link-vrf different from their vrf: their cross-vrf must be vrf main.

---

The first two paragraphs describe how to use GRE interfaces and NHRP to establish connections between the spokes via the hub, either statically or dynamically.

The last paragraph introduces the `redirect` feature that enables dynamic tunnel establishment between spokes.

### Traffic flowing between spoke and hub

Once the GRE interface is set up on the spokes, it is possible to use NHRP to mount routes to reach remote networks from hub.

It is possible to configure NHRP static entries on both hub and spoke sides, so that NHRP routes will be automatically added in the system. Note that this mechanism does not involve any communication between both sides, from NHRP protocol point of view.

The other solution involves a transaction between the spokes and the hub. A spoke configures the NHRP service, and declares a NHS. Actually NHS stands for the hub device. Once done, a spoke begins the transaction by sending a NHRP registration request. This request contains the NBMA IP address associated to the protocol IP address of the GRE interface. For the hub, the reception of incoming requests will lead to a registration reply indicating the success (or the failure) of the operation, along with its NBMA and protocol IP address.

### Traffic flowing between spokes

It is possible to send the traffic directly between spokes, by relying on point to multipoint GRE interfaces on the spokes side too.

Tunnels are dynamically established, based on a `nhrp redirect` feature located on the hub side. NHRP helps on the hub side by identifying traffic that is flowing in and out through the same GRE interface. The hub takes a snapshot and sends some NHRP redirect information to the relevant spokes. At the end, the negotiation finalises between the spokes.

## RFC

**RFC 2332** (<https://tools.ietf.org/html/rfc2332.html>): NBMA Next Hop Resolution Protocol (NHRP)

**RFC 2333** (<https://tools.ietf.org/html/rfc2333.html>): NHRP protocol applicability statement

### See also:

The *NHRP command reference*

### DMVPN and NHRP Configuration

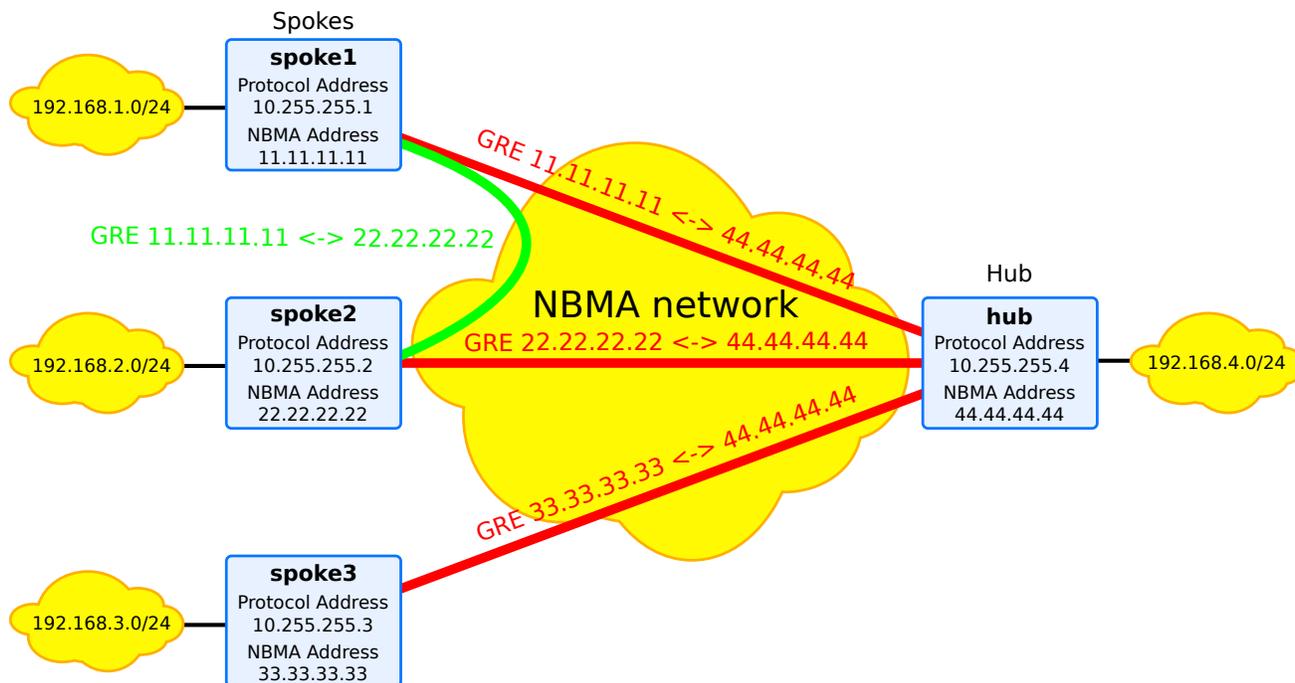


Fig. 13: NHRP use case example

The following configuration examples apply on the topology described in the above picture: 3 spokes connected to a hub. The first two chapters describe how to use GRE interfaces and NHRP to establish connections between the spokes via the hub, either statically or dynamically. The last chapter introduces the `redirect` feature that enables dynamic tunnel establishment between spokes.

For all those examples, NHRP configuration requires that the IP address of GRE interface be configured with 32 bitmask, as follows. Configuring an other bitmask will lead to a misconfiguration problem.

**spoke1**

```

spoke1 running config# vrf main
spoke1 running vrf main# interface gre gre1
spoke1 running gre gre1# ipv4 address 10.255.255.1/32

```

**Static configuration**

We can define static NHRP entries for remote endpoints on each node. From a protocol point of view, hub and spokes don't exchange messages then. The hub declares the protocol address and the NBMA address of the spokes. The spokes only declare a NHRP entry for the hub. In that case, spoke-to-spoke communications flow through the hub. Therefore the hub declares a multipoint GRE interface facing all the spokes and the spokes declare a GRE interface facing the hub only. In addition, each node runs a BGP instance to advertise its private network and learn the route towards the private subnets of the other nodes.

First we configure the three spokes with a static NHRP entry for the hub:

**spoke1**

```

spoke1 running config# vrf main
spoke1 running vrf main# interface physical wan
spoke1 running physical wan#! port pci-b0s4
spoke1 running physical wan# ipv4 address 11.11.11.11/24
spoke1 running physical wan# ..
spoke1 running interface# physical lan
spoke1 running physical lan#! port pci-b0s5
spoke1 running physical lan# ipv4 address 192.168.1.1/24
spoke1 running physical lan# ..
spoke1 running interface# gre gre1
spoke1 running gre gre1#! ipv4 address 10.255.255.1/32
spoke1 running gre gre1#! link-interface wan
spoke1 running gre gre1#! local 11.11.11.11
spoke1 running gre gre1# remote 44.44.44.44
spoke1 running gre gre1# .. ..
spoke1 running vrf main# routing
spoke1 running routing# static ipv4-route 0.0.0.0/0 next-hop 11.11.11.1
spoke1 running routing# nhrp enabled true
spoke1 running routing# interface gre1 ip nhrp
spoke1 running nhrp# registration-no-unique true
spoke1 running nhrp# network-id 123
spoke1 running nhrp# holdtime 1200
spoke1 running nhrp# nhrp-map 10.255.255.4 nbma 44.44.44.44
spoke1 running nhrp# .. .. ..

```

(continues on next page)

(continued from previous page)

```

spoke1 running routing# bgp
spoke1 running bgp#! as 65000
spoke1 running bgp# router-id 10.255.255.1
spoke1 running bgp# address-family ipv4-unicast network 192.168.1.0/24
spoke1 running network 192.168.1.0/24# ..
spoke1 running ipv4-unicast# network 10.255.255.1/32
spoke1 running network 10.255.255.255.1/32# .. .. .
spoke1 running bgp# neighbor 10.255.255.4
spoke1 running neighbor 10.255.255.4#! remote-as 65000
spoke1 running neighbor 10.255.255.4# commit

```

**spoke2**

```

spoke2 running config# vrf main
spoke2 running vrf main# interface physical wan
spoke2 running physical wan#! port pci-b0s4
spoke2 running physical wan# ipv4 address 22.22.22.22/24
spoke2 running physical wan# ..
spoke2 running interface# physical lan
spoke2 running physical lan#! port pci-b0s5
spoke2 running physical lan# ipv4 address 192.168.2.1/24
spoke2 running physical lan# ..
spoke2 running interface# gre gre2
spoke2 running gre gre2#! ipv4 address 10.255.255.2/32
spoke2 running gre gre2#! link-interface wan
spoke2 running gre gre2#! local 22.22.22.22
spoke2 running gre gre2# remote 44.44.44.44
spoke2 running gre gre2# .. ..
spoke2 running vrf main# routing
spoke2 running routing# static ipv4-route 0.0.0.0/0 next-hop 22.22.22.1
spoke2 running routing# nhrp enabled true
spoke2 running routing# interface gre2 ip nhrp
spoke2 running nhrp# registration-no-unique true
spoke2 running nhrp# network-id 123
spoke2 running nhrp# holdtime 1200
spoke2 running nhrp# nhrp-map 10.255.255.4 nbma 44.44.44.44
spoke2 running nhrp# .. .. .
spoke2 running routing# bgp
spoke2 running bgp#! as 65000
spoke2 running bgp# router-id 10.255.255.2
spoke2 running bgp# address-family ipv4-unicast network 192.168.2.0/24
spoke1 running network 192.168.2.0/24# ..

```

(continues on next page)

(continued from previous page)

```

spoke1 running ipv4-unicast# network 10.255.255.2/32
spoke1 running network 10.255.255.255.2/32# .. .. .
spoke2 running bgp# neighbor 10.255.255.4
spoke2 running neighbor 10.255.255.4#! remote-as 65000
spoke2 running neighbor 10.255.255.4# commit

```

### spoke3

```

spoke3 running config# vrf main
spoke3 running vrf main# interface physical wan
spoke3 running physical wan#! port pci-b0s4
spoke3 running physical wan# ipv4 address 33.33.33.33/24
spoke3 running physical wan# ..
spoke3 running interface# physical lan
spoke3 running physical lan#! port pci-b0s5
spoke3 running physical lan# ipv4 address 192.168.3.1/24
spoke3 running physical lan# ..
spoke3 running interface# gre gre3
spoke3 running gre gre3# ipv4 address 10.255.255.3/32
spoke3 running gre gre3# link-interface wan
spoke3 running gre gre3# local 33.33.33.33
spoke3 running gre gre3# remote 44.44.44.44
spoke3 running gre gre3# .. ..
spoke3 running vrf main# routing
spoke3 running routing# static ipv4-route 0.0.0.0/0 next-hop 33.33.33.1
spoke3 running routing# nhrp enabled true
spoke3 running routing# interface gre3 ip nhrp
spoke3 running nhrp# registration-no-unique true
spoke3 running nhrp# network-id 123
spoke3 running nhrp# holdtime 1200
spoke3 running nhrp# nhrp-map 10.255.255.4 nbma 44.44.44.44
spoke3 running nhrp# .. .. .
spoke3 running routing# bgp
spoke3 running bgp#! as 65000
spoke3 running bgp# router-id 10.255.255.3
spoke3 running bgp# address-family ipv4-unicast network 192.168.3.0/24
spoke1 running network 192.168.3.0/24# ..
spoke1 running ipv4-unicast# network 10.255.255.3/32
spoke1 running network 10.255.255.255.3/32# .. .. .
spoke3 running bgp# neighbor 10.255.255.4
spoke3 running neighbor 10.255.255.4#! remote-as 65000
spoke3 running neighbor 10.255.255.4# commit

```

Similarly, we configure static NHRP entries for each spoke on the hub. The main difference is that we configure a multipoint GRE on this node, by omitting a remote address, to be able to reach each spoke:

## hub

```

hub running config# vrf main
hub running vrf main# interface physical wan
hub running physical wan#! port pci-b0s4
hub running physical wan# ipv4 address 44.44.44.44/24
hub running physical wan# ..
hub running interface# physical lan
hub running physical lan#! port pci-b0s5
hub running physical lan# ipv4 address 192.168.4.1/24
hub running physical lan# ..
hub running interface# gre gre4
hub running gre gre4#! ipv4 address 10.255.255.4/32
hub running gre gre4#! link-interface wan
hub running gre gre4#! local 44.44.44.44
hub running gre gre4# .. ..
hub running vrf main# routing
hub running routing# static ipv4-route 0.0.0.0/0 next-hop 44.44.44.1
hub running routing# nhrp enabled true
hub running routing# interface gre4 ip nhrp
hub running nhrp# registration-no-unique true
hub running nhrp# network-id 123
hub running nhrp# holdtime 1200
hub running nhrp# nhrp-map 10.255.255.1 nbma 11.11.11.11
hub running nhrp# nhrp-map 10.255.255.2 nbma 22.22.22.22
hub running nhrp# nhrp-map 10.255.255.3 nbma 33.33.33.33
hub running nhrp# .. .. ..
hub running routing# bgp
hub running bgp#! as 65000
hub running bgp# router-id 10.255.255.4
hub running bgp# address-family ipv4-unicast
hub running ipv4-unicast# network 192.168.4.0/24
hub running network 192.168.4.0/24# .. .. ..
hub running bgp# listen neighbor-range 10.255.255.0/24 neighbor-group nhrp_group
hub running bgp#! neighbor-group nhrp_group
hub running neighbor-group nhrp_group#! remote-as 65000
hub running neighbor-group nhrp_group# address-family ipv4-unicast
hub running ipv4-unicast# nexthop-self force true
hub running ipv4-unicast# route-reflector-client true
hub running ipv4-unicast# commit

```

We can check the NHRP cache entries on each node:

```

hub> show nhrp cache
Iface   Type      Protocol      NBMA           Flags  Identity
gre4    local    10.255.255.4  -              -      -
gre4    static   10.255.255.3  33.33.33.33   -      -
gre4    static   10.255.255.2  22.22.22.22   -      -
gre4    static   10.255.255.1  11.11.11.11   -      -

```

```

spoke1> show nhrp cache
Iface   Type      Protocol      NBMA           Flags  Identity
gre1    static   10.255.255.4  44.44.44.44   -      -
gre1    local    10.255.255.1  -              -      -

```

```

spoke2> show nhrp cache
Iface   Type      Protocol      NBMA           Flags  Identity
gre2    static   10.255.255.4  44.44.44.44   -      -
gre2    local    10.255.255.2  -              -      -

```

```

spoke3> show nhrp cache
Iface   Type      Protocol      NBMA           Flags  Identity
gre3    static   10.255.255.4  44.44.44.44   -      -
gre3    local    10.255.255.3  -              -      -

```

We can see that there is a NHRP connection between the hub and each device:

```

hub> show nhrp-connection
Src          Dst          Flags  SAs  Identity
44.44.44.44  22.22.22.22  n      0
44.44.44.44  33.33.33.33  n      0
44.44.44.44  11.11.11.11  n      0

```

```

spoke1> show nhrp-connection
Src          Dst          Flags  SAs  Identity
11.11.11.11  44.44.44.44  n      0

```

```

spoke2> show nhrp-connection
Src          Dst          Flags  SAs  Identity
22.22.22.22  44.44.44.44  n      0

```

```

spoke3> show nhrp-connection
Src          Dst          Flags  SAs  Identity
33.33.33.33  44.44.44.44  n      0

```

As a consequence, a NHRP route towards the protocol address of the hub is installed on each spoke. This way, they can establish a BGP peering and learn the routes to other nodes:

```

spoke1> show ipv4-routes
[.]
S>* 0.0.0.0/0 [1/0] via 11.11.11.1, wan, 00:01:24
C>* 10.255.255.1/32 is directly connected, gre1, 00:01:24
B> 10.255.255.2/32 [200/0] via 10.255.255.4 (recursive), 00:01:23
*
  via 10.255.255.4, gre1 onlink, 00:01:23
B> 10.255.255.3/32 [200/0] via 10.255.255.4 (recursive), 00:01:23
*
  via 10.255.255.4, gre1 onlink, 00:01:23
N>* 10.255.255.4/32 [10/0] is directly connected, gre1, 00:01:24
C>* 11.11.11.0/24 is directly connected, wan, 00:01:24
C>* 192.168.1.0/24 is directly connected, lan, 00:01:24
B> 192.168.2.0/24 [200/0] via 10.255.255.4 (recursive), 00:01:23
*
  via 10.255.255.4, gre1 onlink, 00:01:23
B> 192.168.3.0/24 [200/0] via 10.255.255.4 (recursive), 00:01:23
*
  via 10.255.255.4, gre1 onlink, 00:01:23
B> 192.168.4.0/24 [200/0] via 10.255.255.4 (recursive), 00:01:23
*
  via 10.255.255.4, gre1 onlink, 00:01:23

```

```

spoke2> show ipv4-routes
[.]
S>* 0.0.0.0/0 [1/0] via 22.22.22.1, wan, 00:07:21
B> 10.255.255.1/32 [200/0] via 10.255.255.4 (recursive), 00:07:20
*
  via 10.255.255.4, gre2 onlink, 00:07:20
C>* 10.255.255.2/32 is directly connected, gre2, 00:07:21
B> 10.255.255.3/32 [200/0] via 10.255.255.4 (recursive), 00:07:20
*
  via 10.255.255.4, gre2 onlink, 00:07:20
N>* 10.255.255.4/32 [10/0] is directly connected, gre2, 00:07:21
C>* 22.22.22.0/24 is directly connected, wan, 00:07:21
B> 192.168.1.0/24 [200/0] via 10.255.255.4 (recursive), 00:07:20
*
  via 10.255.255.4, gre2 onlink, 00:07:20
C>* 192.168.2.0/24 is directly connected, lan, 00:07:21
B> 192.168.3.0/24 [200/0] via 10.255.255.4 (recursive), 00:07:20
*
  via 10.255.255.4, gre2 onlink, 00:07:20
B> 192.168.4.0/24 [200/0] via 10.255.255.4 (recursive), 00:07:20
*
  via 10.255.255.4, gre2 onlink, 00:07:20

```

```

spoke3> show ipv4-routes
[.]
S>* 0.0.0.0/0 [1/0] via 33.33.33.1, wan, 00:08:24
B> 10.255.255.1/32 [200/0] via 10.255.255.4 (recursive), 00:08:22
*
  via 10.255.255.4, gre3 onlink, 00:08:22
B> 10.255.255.2/32 [200/0] via 10.255.255.4 (recursive), 00:08:23
*
  via 10.255.255.4, gre3 onlink, 00:08:23
C>* 10.255.255.3/32 is directly connected, gre3, 00:08:24

```

(continues on next page)

(continued from previous page)

```

N>* 10.255.255.4/32 [10/0] is directly connected, gre3, 00:08:24
C>* 33.33.33.0/24 is directly connected, wan, 00:08:24
B> 192.168.1.0/24 [200/0] via 10.255.255.4 (recursive), 00:08:22
*
   via 10.255.255.4, gre3 onlink, 00:08:22
B> 192.168.2.0/24 [200/0] via 10.255.255.4 (recursive), 00:08:23
*
   via 10.255.255.4, gre3 onlink, 00:08:23
C>* 192.168.3.0/24 is directly connected, lan, 00:08:24
B> 192.168.4.0/24 [200/0] via 10.255.255.4 (recursive), 00:08:23
*
   via 10.255.255.4, gre3 onlink, 00:08:23

```

The hub installs a NHRP route for each spoke and learns the remote private networks through BGP:

```

hub> show ipv4-routes
[.]
S>* 0.0.0.0/0 [1/0] via 44.44.44.1, wan, 00:09:05
B 10.255.255.1/32 [200/0] via 10.255.255.1 inactive, 00:09:04
N>* 10.255.255.1/32 [10/0] is directly connected, gre4, 00:09:05
B 10.255.255.2/32 [200/0] via 10.255.255.2 inactive, 00:09:04
N>* 10.255.255.2/32 [10/0] is directly connected, gre4, 00:09:05
B 10.255.255.3/32 [200/0] via 10.255.255.3 inactive, 00:09:04
N>* 10.255.255.3/32 [10/0] is directly connected, gre4, 00:09:05
C>* 10.255.255.4/32 is directly connected, gre4, 00:09:05
C>* 44.44.44.0/24 is directly connected, wan, 00:09:05
B> 192.168.1.0/24 [200/0] via 10.255.255.1 (recursive), 00:09:04
*
   via 10.255.255.1, gre4 onlink, 00:09:04
B> 192.168.2.0/24 [200/0] via 10.255.255.2 (recursive), 00:09:04
*
   via 10.255.255.2, gre4 onlink, 00:09:04
B> 192.168.3.0/24 [200/0] via 10.255.255.3 (recursive), 00:09:04
*
   via 10.255.255.3, gre4 onlink, 00:09:04
C>* 192.168.4.0/24 is directly connected, lan, 00:09:05

```

## Dynamic configuration

*Static configuration* doesn't scale well as you have to declare each spoke individually on the hub. We can fully leverage NHRP and use dynamic configuration instead. In this configuration, the NHRP server on the hub listens for registration requests. Each spoke declares the hub as a NHRP server and thus registers upon startup using NHRP control packets. With the basic setup below, spoke-to-spoke communication still runs through the hub but can then be improved to allow *Direct spoke-to-spoke communication*. That's why we configure point to multipoint GRE interface on both spoke and hub side.

## BGP configuration

In addition, each node runs a BGP instance to advertise its private network. As the chapter deals with spoke to *hub* communication, only *hub* is interested in getting the private networks located behind spokes.

## iBGP configuration

It is possible to keep using iBGP sessions between spokes and *hub*, like it has been done on previous experiment. In that case, spokes will install iBGP routes to reach private networks from other spokes, but by using *hub* as nextthop. A snapshot of configuration can be illustrated below with *spoke1*, *spoke2* and *hub* devices:

Below dump gives the initial ipv4 routing table on *spoke1*. One can see routes installed to reach *spoke2*, *spoke3* and *hub*.

```
spoke1> show ipv4-routes
[..]
S>* 0.0.0.0/0 [1/0] via 11.11.11.1, wan, 00:01:24
C>* 10.255.255.1/32 is directly connected, gre1, 00:01:24
B> 10.255.255.2/32 [200/0] via 10.255.255.4 (recursive), 00:01:23
*
   via 10.255.255.4, gre1 onlink, 00:01:23
B> 10.255.255.3/32 [200/0] via 10.255.255.4 (recursive), 00:01:23
*
   via 10.255.255.4, gre1 onlink, 00:01:23
N>* 10.255.255.4/32 [10/0] is directly connected, gre1, 00:01:24
C>* 11.11.11.0/24 is directly connected, wan, 00:01:24
C>* 192.168.1.0/24 is directly connected, lan, 00:01:24
B> 192.168.2.0/24 [200/0] via 10.255.255.4 (recursive), 00:01:23
*
   via 10.255.255.4, gre1 onlink, 00:01:23
B> 192.168.3.0/24 [200/0] via 10.255.255.4 (recursive), 00:01:23
*
   via 10.255.255.4, gre1 onlink, 00:01:23
B> 192.168.4.0/24 [200/0] via 10.255.255.4 (recursive), 00:01:23
*
   via 10.255.255.4, gre1 onlink, 00:01:23
```

## eBGP configuration

It is also possible to avoid transmitting the private networks information to all spokes. Two reasons for that:

- In that setup where data is always passing through the hub, there is no point giving this information to spokes that always point to *hub*.
- This routing information is redundant with NHRP when direct spoke to spoke communication will be put in place. Even if iBGP routes were used, NHRP routes to those private networks would override those BGP routes, and NHRP routes would directly have the nextthop set to the appropriate spoke.

For the rest of this chapter, We choose to use eBGP configuration. The *hub* will transmit two aggregated networks, 192.168.0.0/16 and 10.255.255.0/24. This stands for the subnetworks to all spokes. You can note that eBGP

requires peers to be directly connected. As the route to reach the *hub* IP on the spoke is a NHRP route and not a connected route, an additional configuration is put in place, in order to inject incoming BGP network. This is done via `ebgp-connected-route-check` BGP configuration command:

---

**Note:** Using EBGP avoids *spokes* to learn BGP route entries from *spokes* on same AS coming from an external AS. This specificity works because the *spokes* BGP configurations use the same AS and are not fully meshed.

---

**Note:** An other alternative to defining aggregated networks on *hub* side is to define a default-route at *hub* side, like below BGP command shows it:

```
hub running bgp# neighbour-group nhrp_group address-family ipv4-unicast
hub running ipv4-unicast# default-originate
hub running default-originate# ..
hub running ipv4-unicast# commit
```

For that, there must not be any conflicts with learnt default route of the ISPs of local spokes. A good practice would be to build GRE interfaces on separate VRF from wan interface. Below example illustrates how to move NHRP contexts and lan interface to a VRF `nhrp`. In that case, NHRP routes will be visible from the VRF `nhrp`.

```
spoke1 running config# vrf nhrp
spoke1 running vrf nhrp# interface physical lan
spoke1 running physical lan#! port pci-b0s5
spoke1 running physical lan# ipv4 address 192.168.1.1/24
spoke1 running physical lan# ..
spoke1 running interface# gre gre1
spoke1 running gre gre1#! ipv4 address 10.255.255.1/32
spoke1 running gre gre1#! link-interface wan
spoke1 running gre gre1#! link-vrf main
spoke1 running gre gre1#! local 11.11.11.11
spoke1 running gre gre1# ttl 64
spoke1 running gre gre1# .. ..
spoke1 running vrf nhrp# routing
spoke1 running routing# nhrp enabled true
spoke1 running routing# interface gre1 ip nhrp
spoke1 running nhrp# registration-no-unique true
spoke1 running nhrp# network-id 123
spoke1 running nhrp# holdtime 1200
spoke1 running nhrp# nhrp-nhs dynamic nbma 44.44.44.44
spoke1 running nhrp# .. .. ..
spoke1 running routing# bgp
spoke1 running bgp#! as 65099
spoke1 running bgp# ebgp-connected-route-check false
spoke1 running bgp# router-id 10.255.255.1
```

(continues on next page)

(continued from previous page)

```

spoke1 running bgp# address-family ipv4-unicast network 192.168.1.0/24
spoke1 running network 192.168.1.0/24# .. .. .
spoke1 running bgp# neighbor 10.255.255.4
spoke1 running neighbor 10.255.255.4#! remote-as 65000
spoke1 running neighbor 10.255.255.4# commit

```

## Configuration

The following configuration applies on the same topology with one hub and three spokes.

### spoke1

```

spoke1 running config# vrf main
spoke1 running vrf main# interface physical wan
spoke1 running physical wan#! port pci-b0s4
spoke1 running physical wan# ipv4 address 11.11.11.11/24
spoke1 running physical wan# ..
spoke1 running interface# physical lan
spoke1 running physical lan#! port pci-b0s5
spoke1 running physical lan# ipv4 address 192.168.1.1/24
spoke1 running physical lan# ..
spoke1 running interface# gre gre1
spoke1 running gre gre1#! ipv4 address 10.255.255.1/32
spoke1 running gre gre1#! link-interface wan
spoke1 running gre gre1#! local 11.11.11.11
spoke1 running gre gre1#! ttl 64
spoke1 running gre gre1# .. ..
spoke1 running vrf main# routing
spoke1 running routing# static ipv4-route 0.0.0.0/0 next-hop 11.11.11.1
spoke1 running routing# nhrp enabled true
spoke1 running routing# interface gre1 ip nhrp
spoke1 running nhrp# registration-no-unique true
spoke1 running nhrp# network-id 123
spoke1 running nhrp# holdtime 1200
spoke1 running nhrp# nhrp-nhs dynamic nbma 44.44.44.44
spoke1 running nhrp# .. .. .
spoke1 running routing# bgp
spoke1 running bgp#! as 65099
spoke1 running bgp# ebgp-connected-route-check false
spoke1 running bgp# router-id 10.255.255.1

```

(continues on next page)

(continued from previous page)

```
spoke1 running bgp# address-family ipv4-unicast network 192.168.1.0/24
spoke1 running network 192.168.1.0/24# .. .. .
spoke1 running bgp# neighbor 10.255.255.4
spoke1 running neighbor 10.255.255.4#! remote-as 65000
spoke1 running neighbor 10.255.255.4# commit
```

## spoke2

```
spoke2 running config# vrf main
spoke2 running vrf main# interface physical wan
spoke2 running physical wan#! port pci-b0s4
spoke2 running physical wan# ipv4 address 22.22.22.22/24
spoke2 running physical wan# ..
spoke2 running interface# physical lan
spoke2 running physical lan#! port pci-b0s5
spoke2 running physical lan# ipv4 address 192.168.2.1/24
spoke2 running physical lan# ..
spoke2 running interface# gre gre2
spoke2 running gre gre2#! ipv4 address 10.255.255.2/32
spoke2 running gre gre2#! link-interface wan
spoke2 running gre gre2#! local 22.22.22.22
spoke2 running gre gre2# ttl 64
spoke2 running gre gre2# .. ..
spoke2 running vrf main# routing
spoke2 running routing# static ipv4-route 0.0.0.0/0 next-hop 22.22.22.1
spoke2 running routing# nhrp enabled true
spoke2 running routing# interface gre2 ip nhrp
spoke2 running nhrp# registration-no-unique true
spoke2 running nhrp# network-id 123
spoke2 running nhrp# holdtime 1200
spoke2 running nhrp# nhrp-nhs dynamic nbma 44.44.44.44
spoke2 running nhrp# .. .. .
spoke2 running routing# bgp
spoke2 running bgp#! as 65099
spoke2 running bgp# ebgp-connected-route-check false
spoke2 running bgp# router-id 10.255.255.2
spoke2 running bgp# address-family ipv4-unicast network 192.168.2.0/24
spoke2 running network 192.168.2.0/24# .. .. .
spoke2 running bgp# neighbor 10.255.255.4
spoke2 running neighbor 10.255.255.4#! remote-as 65000
spoke2 running neighbor 10.255.255.4# commit
```

**spoke3**

```

spoke3 running config# vrf main
spoke3 running vrf main# interface physical wan
spoke3 running physical wan#! port pci-b0s4
spoke3 running physical wan# ipv4 address 33.33.33.33/24
spoke3 running physical wan# ..
spoke3 running interface# physical lan
spoke3 running physical lan#! port pci-b0s5
spoke3 running physical lan# ipv4 address 192.168.3.1/24
spoke3 running physical lan# ..
spoke3 running interface# gre gre3
spoke3 running gre gre3#! ipv4 address 10.255.255.3/32
spoke3 running gre gre3#! link-interface wan
spoke3 running gre gre3#! local 33.33.33.33
spoke3 running gre gre3# ttl 64
spoke3 running gre gre3# .. ..
spoke3 running vrf main# routing
spoke3 running routing# static ipv4-route 0.0.0.0/0 next-hop 33.33.33.1
spoke3 running routing# nhrp enabled true
spoke3 running routing# interface gre3 ip nhrp
spoke3 running nhrp# registration-no-unique true
spoke3 running nhrp# network-id 123
spoke3 running nhrp# holdtime 1200
spoke3 running nhrp# nhrp-nhs dynamic nbma 44.44.44.44
spoke3 running nhrp# .. .. ..
spoke3 running routing# bgp
spoke3 running bgp#! as 65099
spoke3 running bgp# ebgp-connected-route-check false
spoke3 running bgp# router-id 10.255.255.3
spoke3 running bgp# address-family ipv4-unicast network 192.168.3.0/24
spoke3 running network 192.168.3.0/24# .. .. ..
spoke3 running bgp# neighbor 10.255.255.4
spoke3 running neighbor 10.255.255.4#! remote-as 65000
spoke3 running neighbor 10.255.255.4# commit

```

Similarly, we do not declare any spoke entry on the hub. This node only listens for NHRP registration requests and then for BGP peerings coming from the spokes.

**hub**

```

hub running config# vrf main
hub running vrf main# interface physical wan
hub running physical wan#! port pci-b0s4
hub running physical wan# ipv4 address 44.44.44.44/24
hub running physical wan# ..
hub running interface# physical lan
hub running physical lan#! port pci-b0s5
hub running physical lan# ipv4 address 192.168.4.1/24
hub running physical lan# ..
hub running interface# gre gre4
hub running gre gre4#! ipv4 address 10.255.255.4/32
hub running gre gre4#! link-interface wan
hub running gre gre4#! local 44.44.44.44
hub running gre gre1# ttl 64
hub running gre gre4# .. ..
hub running vrf main# routing
hub running routing# static ipv4-route 0.0.0.0/0 next-hop 44.44.44.1
hub running routing# nhrp enabled true
hub running routing# interface gre4 ip nhrp
hub running nhrp# registration-no-unique true
hub running nhrp# network-id 123
hub running nhrp# holdtime 1200
hub running nhrp# .. .. ..
hub running routing# bgp
hub running bgp#! as 65000
hub running bgp# router-id 10.255.255.4
hub running bgp# ebgp-connected-route-check false
hub running bgp# listen neighbor-range 10.255.255.0/24 neighbor-group nhrp_group
hub running bgp#! neighbor-group nhrp_group
hub running neighbor-group nhrp_group#! remote-as 65099
hub running neighbor-group nhrp_group# ..
hub running bgp# address-family ipv4-unicast
hub running ipv4-unicast# network 10.255.255.0/24
hub running network 10.255.255.255.0/24# ..
hub running ipv4-unicast# network 192.168.0.0/16
hub running network 192.168.0.0/16# commit

```

We can check the NHRP cache entries on each node:

```

hub> show nhrp cache

```

| Iface | Type    | Protocol     | NBMA        | Flags | Identity |
|-------|---------|--------------|-------------|-------|----------|
| gre4  | local   | 10.255.255.4 | -           |       | -        |
| gre4  | dynamic | 10.255.255.3 | 33.33.33.33 | T     |          |

(continues on next page)

(continued from previous page)

|      |         |              |             |   |
|------|---------|--------------|-------------|---|
| gre4 | dynamic | 10.255.255.2 | 22.22.22.22 | T |
| gre4 | dynamic | 10.255.255.1 | 11.11.11.11 | T |

```
spoke1> show nhrp cache
```

| Iface | Type  | Protocol     | NBMA        | Flags | Identity |
|-------|-------|--------------|-------------|-------|----------|
| gre1  | nhs   | 10.255.255.4 | 44.44.44.44 | T     |          |
| gre1  | local | 10.255.255.1 | -           |       | -        |

```
spoke2> show nhrp cache
```

| Iface | Type  | Protocol     | NBMA        | Flags | Identity |
|-------|-------|--------------|-------------|-------|----------|
| gre2  | nhs   | 10.255.255.4 | 44.44.44.44 | T     |          |
| gre2  | local | 10.255.255.2 | -           |       | -        |

```
spoke3> show nhrp cache
```

| Iface | Type  | Protocol     | NBMA        | Flags | Identity |
|-------|-------|--------------|-------------|-------|----------|
| gre3  | nhs   | 10.255.255.4 | 44.44.44.44 | T     |          |
| gre3  | local | 10.255.255.3 | -           |       | -        |

And the NHRP connections: each spoke has an NHRP connection with the hub:

```
hub> show nhrp-connection
```

| Src         | Dst         | Flags | SAs | Identity |
|-------------|-------------|-------|-----|----------|
| 44.44.44.44 | 22.22.22.22 | n     | 0   |          |
| 44.44.44.44 | 33.33.33.33 | n     | 0   |          |
| 44.44.44.44 | 11.11.11.11 | n     | 0   |          |

```
spoke1> show nhrp-connection
```

| Src         | Dst         | Flags | SAs | Identity |
|-------------|-------------|-------|-----|----------|
| 11.11.11.11 | 44.44.44.44 | n     | 0   |          |

```
spoke2> show nhrp-connection
```

| Src         | Dst         | Flags | SAs | Identity |
|-------------|-------------|-------|-----|----------|
| 22.22.22.22 | 44.44.44.44 | n     | 0   |          |

```
spoke3> show nhrp-connection
```

| Src         | Dst         | Flags | SAs | Identity |
|-------------|-------------|-------|-----|----------|
| 33.33.33.33 | 44.44.44.44 | n     | 0   |          |

We can also dump the link layer contexts of gre4 interface on the hub, to check the status of connected spokes, by using following command:

```
hub> show state / vrf main interface gre gre4 ipv4
ipv4
```

(continues on next page)

(continued from previous page)

```

address 10.255.255.4/32
neighbor 10.255.255.2 link-layer-address 22.22.22.22 state reachable
neighbor 10.255.255.3 link-layer-address 33.33.33.33 state reachable
neighbor 10.255.255.1 link-layer-address 11.11.11.11 state reachable

```

As a consequence, a NHRP route towards the protocol address of the hub is installed on each spoke. This way, each spoke can establish a BGP peering session and learn the routes to the *hub*:

```

spoke1> show ipv4-routes
[.]
S>* 0.0.0.0/0 [1/0] via 11.11.11.1, wan, 00:01:59
B> 10.255.255.0/24 [20/0] via 10.255.255.4 (recursive), 00:01:58
*
   via 10.255.255.4, gre1 onlink, 00:01:58
C>* 10.255.255.1/32 is directly connected, gre1, 00:01:59
N>* 10.255.255.4/32 [10/0] is directly connected, gre1, 00:01:59
C>* 11.11.11.0/24 is directly connected, wan, 00:01:59
B> 192.168.0.0/16 [20/0] via 10.255.255.4 (recursive), 00:01:58
*
   via 10.255.255.4, gre1 onlink, 00:01:58
C>* 192.168.1.0/24 is directly connected, lan, 00:01:59

```

```

spoke2> show ipv4-routes
[.]
S>* 0.0.0.0/0 [1/0] via 22.22.22.1, wan, 00:03:28
B> 10.255.255.0/24 [20/0] via 10.255.255.4 (recursive), 00:03:27
*
   via 10.255.255.4, gre2 onlink, 00:03:27
C>* 10.255.255.2/32 is directly connected, gre2, 00:03:28
N>* 10.255.255.4/32 [10/0] is directly connected, gre2, 00:03:28
C>* 22.22.22.0/24 is directly connected, wan, 00:03:28
B> 192.168.0.0/16 [20/0] via 10.255.255.4 (recursive), 00:03:27
*
   via 10.255.255.4, gre2 onlink, 00:03:27
C>* 192.168.2.0/24 is directly connected, lan, 00:03:28

```

```

spoke3> show ipv4-routes
[.]
S>* 0.0.0.0/0 [1/0] via 33.33.33.1, wan, 00:04:25
B> 10.255.255.0/24 [20/0] via 10.255.255.4 (recursive), 00:04:24
*
   via 10.255.255.4, gre3 onlink, 00:04:24
C>* 10.255.255.3/32 is directly connected, gre3, 00:04:25
N>* 10.255.255.4/32 [10/0] is directly connected, gre3, 00:04:25
C>* 33.33.33.0/24 is directly connected, wan, 00:04:25
B> 192.168.0.0/16 [20/0] via 10.255.255.4 (recursive), 00:04:24
*
   via 10.255.255.4, gre3 onlink, 00:04:24
C>* 192.168.3.0/24 is directly connected, lan, 00:04:25

```

The hub installs a NHRP route for each spoke and learns the remote private networks through BGP:

```

hub> show ipv4-routes
[.]
S>* 0.0.0.0/0 [1/0] via 44.44.44.1, wan, 00:06:28
N>* 10.255.255.1/32 [10/0] is directly connected, gre4, 00:06:27
N>* 10.255.255.2/32 [10/0] is directly connected, gre4, 00:06:27
N>* 10.255.255.3/32 [10/0] is directly connected, gre4, 00:06:27
C>* 10.255.255.4/32 is directly connected, gre4, 00:06:28
C>* 44.44.44.0/24 is directly connected, wan, 00:06:28
B> 192.168.1.0/24 [20/0] via 10.255.255.1 (recursive), 00:06:26
*
* via 10.255.255.1, gre4 onlink, 00:06:26
B> 192.168.2.0/24 [20/0] via 10.255.255.2 (recursive), 00:06:26
*
* via 10.255.255.2, gre4 onlink, 00:06:26
B> 192.168.3.0/24 [20/0] via 10.255.255.3 (recursive), 00:06:26
*
* via 10.255.255.3, gre4 onlink, 00:06:26
C>* 192.168.4.0/24 is directly connected, lan, 00:06:28

```

As you can see, the remote networks are learnt via BGP server from hub, and NHRP routes are set up with a higher priority (10) compared with BGP (20).

All traffic goes through the hub device. This is also the main drawback of this configuration, since it is not possible to establish spoke-to-spoke traffic.

### Direct spoke-to-spoke communication

This enhancement enables direct spoke-to-spoke traffic, without routing through the hub. It uses the NHRP redirect feature and must be applied on top of a *Dynamic configuration*.

This feature, once enabled on the hub, detects a traffic flow that enters and exits the hub through the same GRE interface. This traffic flow can be redirected as the hub is only routing the flow between two spokes. Then the NHRP daemon of the hub issues a NHRP traffic indication message containing the detected packet to the emitter. A spoke receiving such a message performs a NHRP resolution request to find the NBMA address of the destination and then continues the communication directly with the destination spoke.

---

**Note:** The traffic flow detection mechanism is implemented by the fast path. Therefore, the fast path must be enabled on the hub to benefit from the spoke-to-spoke direct communication.

---

We first enable the redirect feature on the hub:

```

hub> edit running
hub running config# / vrf main routing interface gre4 ip nhrp redirect true
hub running config#! / vrf main routing nhrp hub-mode true
hub running config# commit

```

This enables the sampling of the overlay traffic of the gre4 interface. Only the packets that come from this interface and are forwarded through the same interface are eligible for capture. In order to avoid exhausting the device CPU, the NHRP daemon applies a default sampling rate of 1 packet out of 400. If the destination address matches one of the registered spokes, the hub attaches the packet to a NHRP traffic indication packet, and sends it to the sender spoke.

Then we enable the shortcut feature on the spokes so that these messages get processed:

```
spoke1> edit running
spoke1 running config# vrf main routing interface gre1 ip nhrp shortcut true
spoke1 running config# commit
```

```
spoke2> edit running
spoke2 running config# vrf main routing interface gre2 ip nhrp shortcut true
spoke2 running config# commit
```

```
spoke3> edit running
spoke3 running config# vrf main routing interface gre3 ip nhrp shortcut true
spoke3 running config# commit
```

As the message contains the original destination IP, upon reception, the spoke sends a NHRP resolution request, routed by the hub and replied by the destination spoke. Then both spokes continue the communication bypassing the hub.

On direct spoke to spoke communication, one can distinguish two kinds of traffic. Each traffic will result in NHRP messages exchange involving spokes and *hub*, as depicted above.

- The protocol address to protocol address communication between spokes. This is done by issuing traffic between those IP addresses. It will result in the creation of a dynamic cache entry.
- The traffic between private networks located behind GRE interfaces. In addition to the creation of the above mentioned dynamic cache entry, it will result in the creation of a shortcut entry.

## Dynamic cache entry

Let's perform traffic between protocol address of each spoke. To achieve this communication, NHRP protocol will create a dynamic cache entry identifying the two spokes that want to communicate together. From dataplane perspective, a NHRP connected route will be setup on the GRE interface, after the mutual resolution. Example is given below by issuing following cmd ping command:

```
spoke1 running config# cmd ping 10.255.255.2 source 10.255.255.1 vrf main rate 100
```

The cache entries will look like below:

```
spoke1> show nhrp cache
Iface      Type      Protocol      NBMA      Flags      Identity
```

(continues on next page)

(continued from previous page)

|      |         |              |             |   |   |
|------|---------|--------------|-------------|---|---|
| gre1 | nhs     | 10.255.255.4 | 44.44.44.44 | T |   |
| gre1 | dynamic | 10.255.255.2 | 22.22.22.22 | T |   |
| gre1 | local   | 10.255.255.1 | -           |   | - |

```
spoke2> show nhrp cache
```

| Iface | Type    | Protocol     | NBMA        | Flags | Identity |
|-------|---------|--------------|-------------|-------|----------|
| gre2  | nhs     | 10.255.255.4 | 44.44.44.44 | T     |          |
| gre2  | local   | 10.255.255.2 | -           |       | -        |
| gre2  | dynamic | 10.255.255.1 | 44.44.44.44 | T     |          |

It is worth to be noted that in the last `spoke2` dump, NBMA address can be the *hub* NBMA address, because the resolution request coming from `spoke1` has been forwarded by *hub*, and no direct resolution request has been received by `spoke2` from `spoke1`. To be sure that the association between `spoke1` protocol address and its NBMA address has been correctly done, the attribute `NBMA-NAT-OA-Address` in `show nhrp` command indicates the real NBMA address used.

```
spoke2> show nhrp
[.]
Type: dynamic
Protocol-Address: 10.255.255.1/32
NBMA-Address: 44.44.44.44
NBMA-NAT-OA-Address: 11.11.11.11
```

We can see that there is now an additional NHRP connection from `spoke1` to `spoke2`:

```
hub> show nhrp-connection
```

| Src         | Dst         | Flags | SAs | Identity |
|-------------|-------------|-------|-----|----------|
| 44.44.44.44 | 22.22.22.22 | n     | 0   |          |
| 44.44.44.44 | 33.33.33.33 | n     | 0   |          |
| 44.44.44.44 | 11.11.11.11 | n     | 0   |          |

```
spoke1> show nhrp-connection
```

| Src         | Dst         | Flags | SAs | Identity |
|-------------|-------------|-------|-----|----------|
| 11.11.11.11 | 22.22.22.22 | n     | 0   |          |
| 11.11.11.11 | 44.44.44.44 | n     | 0   |          |

Like it has been noted for `spoke2` with `show nhrp cache` dump, the remote entry to 11.11.11.11 may not be present, while no direct resolution has been identified by `spoke2`.

```
spoke2> show nhrp-connection
```

| Src         | Dst         | Flags | SAs | Identity |
|-------------|-------------|-------|-----|----------|
| 22.22.22.22 | 44.44.44.44 | n     | 0   |          |
| 22.22.22.22 | 11.11.11.11 | n     | 0   |          |

As indicated below, a new NHRP connected route entry has been set up. Note that up to now, only traffic between

*Protocol Address* of spokes has been initiated; and that traffic between private networks (192.168.x.0/24) still uses the *hub*.

```
spoke1> show ipv4-routes
[.]
S>* 0.0.0.0/0 [1/0] via 11.11.11.1, wan, 00:45:01
B> 10.255.255.0/24 [20/0] via 10.255.255.4 (recursive), 00:45:00
*
   via 10.255.255.4, gre1 onlink, 00:45:00
C>* 10.255.255.1/32 is directly connected, gre1, 00:45:01
N>* 10.255.255.2/32 [10/0] is directly connected, gre1, 00:12:14
N>* 10.255.255.4/32 [10/0] is directly connected, gre1, 00:45:01
C>* 11.11.11.0/24 is directly connected, wan, 00:45:01
B> 192.168.0.0/16 [20/0] via 10.255.255.4 (recursive), 00:45:00
*
   via 10.255.255.4, gre1 onlink, 00:45:00
C>* 192.168.1.0/24 is directly connected, lan, 00:45:01
```

```
spoke2> show ipv4-routes
[.]
S>* 0.0.0.0/0 [1/0] via 22.22.22.1, wan, 00:45:38
B> 10.255.255.0/24 [20/0] via 10.255.255.4 (recursive), 00:45:37
*
   via 10.255.255.4, gre2 onlink, 00:45:37
N>* 10.255.255.1/32 [10/0] is directly connected, gre2, 00:12:51
C>* 10.255.255.2/32 is directly connected, gre2, 00:45:38
N>* 10.255.255.4/32 [10/0] is directly connected, gre2, 00:45:38
C>* 22.22.22.0/24 is directly connected, wan, 00:45:38
B> 192.168.0.0/16 [20/0] via 10.255.255.4 (recursive), 00:45:37
*
   via 10.255.255.4, gre2 onlink, 00:45:37
C>* 192.168.2.0/24 is directly connected, lan, 00:45:38
```

## Shortcut entry

Now, let us perform a `cmd ping` command between private networks. This command will result in the capture of this traffic on the *hub* and will lead to the creation of a shortcut entry on spokes.

```
spoke1 running config# cmd ping 192.168.2.1 source 192.168.1.1 vrf main rate 100
```

The NHRP entries help in creating route entries linked to private networks behind each GRE device. For instance, sub-network 192.168.2.0/24 is discovered by NHRP, on *spoke1*, when traffic is issued from *spoke1* to that destination network. On *hub*, the traffic is intercepted and sent to *spoke2* for NHRP resolution. *spoke2* parses the available networks hidden behind GRE interface, and if present, transmits a resolution response to *spoke1* which creates a nexthop route entry linked to *spoke2* dynamic entry, as follows:

```
spoke1> show nhrp shortcut
Type      Prefix                Via                    Identity
```

(continues on next page)

(continued from previous page)

```
dynamic 192.168.2.0/24          10.255.255.2

spoke1> show ipv4-routes
[.]
S>* 0.0.0.0/0 [1/0] via 11.11.11.1, wan, 00:47:58
B> 10.255.255.0/24 [20/0] via 10.255.255.4 (recursive), 00:47:57
*
   via 10.255.255.4, gre1 onlink, 00:47:57
C>* 10.255.255.1/32 is directly connected, gre1, 00:47:58
N>* 10.255.255.2/32 [10/0] is directly connected, gre1, 00:00:48
N>* 10.255.255.4/32 [10/0] is directly connected, gre1, 00:47:58
C>* 11.11.11.0/24 is directly connected, wan, 00:47:58
B> 192.168.0.0/16 [20/0] via 10.255.255.4 (recursive), 00:47:57
*
   via 10.255.255.4, gre1 onlink, 00:47:57
C>* 192.168.1.0/24 is directly connected, lan, 00:47:58
N>* 192.168.2.0/24 [10/0] via 10.255.255.2, gre1 onlink, 00:00:34
```

```
spoke2> show nhrp shortcut
Type      Prefix                Via                    Identity
dynamic 192.168.1.0/24        10.255.255.1

spoke2> show ipv4-routes
[.]
VRF nhrp:
S>* 0.0.0.0/0 [1/0] via 22.22.22.1, wan, 00:49:39
B> 10.255.255.0/24 [20/0] via 10.255.255.4 (recursive), 00:49:38
*
   via 10.255.255.4, gre2 onlink, 00:49:38
N>* 10.255.255.1/32 [10/0] is directly connected, gre2, 00:02:15
C>* 10.255.255.2/32 is directly connected, gre2, 00:49:39
N>* 10.255.255.4/32 [10/0] is directly connected, gre2, 00:49:39
C>* 22.22.22.0/24 is directly connected, wan, 00:49:39
B> 192.168.0.0/16 [20/0] via 10.255.255.4 (recursive), 00:49:38
*
   via 10.255.255.4, gre2 onlink, 00:49:38
N>* 192.168.1.0/24 [10/0] via 10.255.255.1, gre2 onlink, 00:02:34
C>* 192.168.2.0/24 is directly connected, lan, 00:49:39
```

As can be seen, a NHRP onlink route has been created, and reflects the NHRP shortcut entry created. This entry relies on the NHRP dynamic cache entry previously created when initiating traffic between *Protocol Address* of spokes.

---

**Note:** To bring more clarity, the chapter presented how the cache entry is created, then how the shortcut entry is created, according to each incoming traffic. This said, in reality, launching private network traffic will result in the creation of the two NHRP routes necessary to make spoke-to-spoke communication.

---

Nexthop route to 192.168.2.0/24 is introduced, and partially overrides 192.168.0.0/16 defined by BGP. As illustrated, the routing decision in NHRP is made thanks to routing information the protocol has when receiving a resolution request. The traffic indication was about a specific IP whereas the resulting route reflects the route of the remote spoke. In our case, the network 192.168.2.0/24 network information is sent from spoke2 to spoke1. All routing information contained in the vrf where GRE interface sits can be used. A design recommendation when setting up NHRP network would be to isolate GRE inner network in a specific VRF.

---

**Note:** Securing DMVPN connections with IPsec requires a Turbo IPsec Application License.

---

### DMVPN and NHRP security

#### Securing DPVN connections with IPsec

Securing a DMVPN connection requires to configure an IKE VPN. More information on how to configure IKE is given in *IKE user guide*.

A VPN is defined in the IKE context, that requires to encrypt GRE traffic in IPSEC transport mode, and specifies the necessary IKE and IPSEC settings.

The local and remote addresses of the VPN are left unspecified, they will be dynamically provided by the NHRP layer.

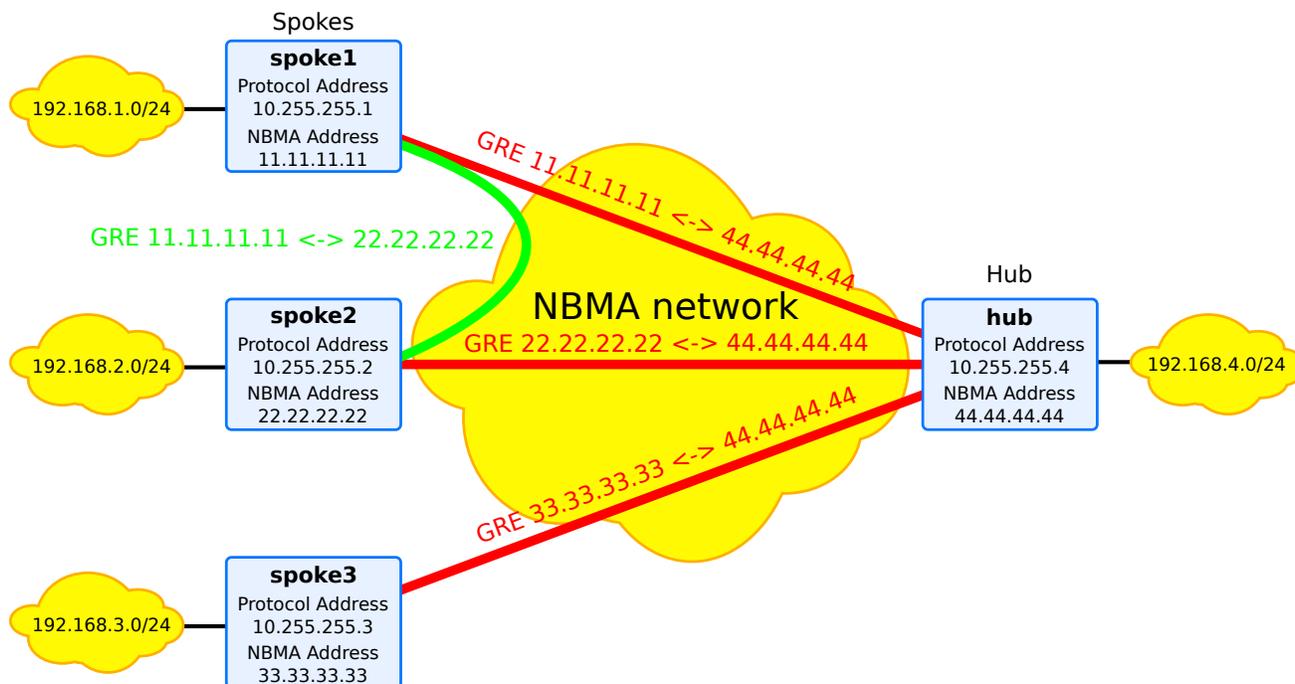


Fig. 14: NHRP use case example

We will now configure all devices of the DMVPN network to encrypt GRE encapsulated traffic with IPSEC.

The procedure consists in configuring an IKE VPN with a `security-policy` for GRE traffic, then to request that the NHRP connection uses this `security-policy` to protect the GRE tunnels (NHRP and data traffic).

### create the IKE VPN

The below configuration defines a VPN with a `security-policy` named `dmvpn-gre`. It has the IKE identity `spoke1`. Each device must have a distinct identity. For example, `hub` identity would be `hub`.

Each instance that wants to secure its connection has to set up similar IKE settings. Basically, only the VPN `local-id` will change.

### spoke1

```
spoke1 running# vrf main
spoke1 running vrf main# ike
spoke1 running ike# pre-shared-key dmvpn-psk
spoke1 running pre-shared-key dmvpn-psk# secret 0seaJ31RfzHNRvUSH0oUYg7znTW0I=
spoke1 running pre-shared-key dmvpn-psk# ..
spoke1 running ike# ike-policy-template ikepol
spoke1 running ike-policy-template ikepol# ike-proposal 1
spoke1 running ike-proposal 1# enc-alg aes256-cbc
spoke1 running ike-proposal 1# auth-alg hmac-sha512
spoke1 running ike-proposal 1# dh-group ecp384
spoke1 running ike-proposal 1# ..
spoke1 running ike-policy-template ikepol# dpd-delay 15
spoke1 running ike-policy-template ikepol# ..
spoke1 running ike# ipsec-policy-template ipsecpol
spoke1 running ipsec-policy-template ipsecpol# esp-proposal 1
spoke1 running esp-proposal 1# enc-alg aes256-cbc
spoke1 running esp-proposal 1# auth-alg hmac-sha512
spoke1 running esp-proposal 1# dh-group ecp384
spoke1 running esp-proposal 1# ..
spoke1 running ipsec-policy-template ipsecpol# start-action none
spoke1 running ipsec-policy-template ipsecpol# close-action none
spoke1 running ipsec-policy-template ipsecpol# dpd-action clear
spoke1 running ipsec-policy-template ipsecpol# rekey-time 100m
spoke1 running ipsec-policy-template ipsecpol# ..
spoke1 running ike# vpn dmvpn
spoke1 running vpn dmvpn# ike-policy
spoke1 running ike-policy# template ikepol
spoke1 running ike-policy# ..
spoke1 running vpn dmvpn# ipsec-policy
```

(continues on next page)

(continued from previous page)

```

spoke1 running ipsec-policy# template ipsecpol
spoke1 running ipsec-policy# ..
spoke1 running vpn dmvpn# local-id spoke1
spoke1 running vpn dmvpn# security-policy dmvpn-gre
spoke1 running security-policy gretunnel# local-ts protocol 47
spoke1 running security-policy gretunnel# remote-ts protocol 47
spoke1 running security-policy gretunnel# mode transport
spoke1 running security-policy gretunnel# ..
spoke1 running vpn dmvpn# ..
spoke1 running ike# ..
spoke1 running vrf main# commit

```

The same configuration can be applied to other *spokes* and *hub*. However, ensure that each device has its own local-id.

### spoke2

```

[.]
spoke2 running# vrf main
spoke2 running vrf main# ike
spoke2 running ike# vpn dmvpn
spoke2 running vpn dmvpn# local-id spoke2
spoke2 running vpn dmvpn# commit

```

### spoke3

```

[.]
spoke3 running# vrf main
spoke3 running vrf main# ike
spoke3 running ike# vpn dmvpn
spoke3 running vpn dmvpn# local-id spoke3
spoke3 running vpn dmvpn# commit

```

### hub

```

[.]
hub running# vrf main
hub running vrf main# ike
hub running ike# vpn dmvpn

```

(continues on next page)

(continued from previous page)

```
hub running vpn dmvpn# local-id hub
hub running vpn dmvpn# commit
```

### reference the IKE VPN in NHRP

Then, the NHRP configuration specifies that the NHRP connection must be protected by an IPsec security-policy named `dmvpn-gre`. The name of the `ipsec-profile` must match the name of the security-policy.

#### spoke1

```
spoke1 running config# vrf main
spoke1 running vrf main# routing interface gre1
spoke1 running interface gre1# nhrp-connection ipsec-profile dmvpn-gre
spoke1 running interface gre1# commit
```

#### spoke2

```
spoke2 running config# vrf main
spoke2 running vrf main# routing interface gre2
spoke2 running interface gre2# nhrp-connection ipsec-profile dmvpn-gre
spoke2 running interface gre2# commit
```

#### spoke3

```
spoke3 running config# vrf main
spoke3 running vrf main# routing interface gre3
spoke3 running interface gre3# nhrp-connection ipsec-profile dmvpn-gre
spoke3 running interface gre3# commit
```

#### hub

```
hub running config# vrf main
hub running vrf main# routing interface gre4
hub running interface gre4# nhrp-connection ipsec-profile dmvpn-gre
hub running interface gre4# commit
```

## IPsec establishment

Thanks to this configuration, prior to sending NHRP packets, the NHRP layer on the spokes will trigger an IKE negotiation between the NBMA addresses of the spoke and the hub, and request the GRE traffic between these addresses to be encrypted in IPSEC transport mode.

Only then the NHRP packets may be exchanged. Both NHRP and data traffic sent through the GRE tunnels will be encrypted by IPSEC.

The command below displays the established IKE SA (Security Association) and their installed child SAs.

### spoke1

```
spoke1> show ike ike-sa vpn dmvpn details
dmvpn: #1, ESTABLISHED, IKEv2, 82fd942f9fdc4325_i 8901f24b124cbe9c_r
  local 'spoke1' @ 11.11.11.11[500]
  remote 'hub' @ 44.44.44.44[500]
  aes256-cbc/hmac-sha512/hmac-sha512/ecp384
  established 714s ago, rekeying in 9499s
dmvpn-gre: #1, reqid 1, INSTALLED, TRANSPORT, esp:aes256-cbc/hmac-sha512
  installed 714s ago, rekeying in 5140s, expires in 5886s
  in c481a614, 106076 bytes, 577 packets
  out cb8a052d, 16032 bytes, 100 packets
  local 11.11.11.11/32
  remote 44.44.44.44/32
```

We can now verify that the NHRP connections are protected by IPSEC. As can be seen, the SAs column stands for the number of child SA used. Identity is the IKE id of the peer.

### hub

```
hub> show nhrp-connection
```

| Src         | Dst         | Flags | SAs | Identity |
|-------------|-------------|-------|-----|----------|
| 44.44.44.44 | 22.22.22.22 | n     | 1   | spoke2   |
| 44.44.44.44 | 33.33.33.33 | n     | 1   | spoke3   |
| 44.44.44.44 | 11.11.11.11 | n     | 1   | spoke1   |

**spoke1**

```
spoke1> show nhrp-connection
Src                Dst                Flags  SAs  Identity
11.11.11.11       44.44.44.44       n      1    hub
```

The same processing occurs between spokes before establishing shortcuts. If the *hub* and *spokes* are set up to allow direct spoke-to-spoke communication, a spoke that receives a traffic indication from the hub will trigger an IKE negotiation with the other spoke, in order to encrypt the GRE traffic between the NBMA addresses of the spokes. Only then the spoke-to-spoke NHRP exchanges may start. The spoke-to-spoke data traffic will also be protected by IPSEC.

**spoke1**

```
spoke1> show nhrp-connection
Src                Dst                Flags  SAs  Identity
11.11.11.11       22.22.22.22       n      1    spoke2
11.11.11.11       44.44.44.44       n      1    hub

spoke1> show ike ike-sa vpn dmvpn details
dmvpn: #2, ESTABLISHED, IKEv2, 633207c251b7df62_i b4dbd7645d4979f2_r
  local  'spoke1' @ 11.11.11.11[500]
  remote 'spoke2' @ 22.22.22.22[500]
  aes256-cbc/hmac-sha512/hmac-sha512/ecp384
  established 420s ago, rekeying in 13606s
  dmvpn-gre: #2, reqid 2, INSTALLED, TRANSPORT, esp:aes256-cbc/hmac-sha512
    installed 420s ago, rekeying in 5345s, expires in 6180s
    in  c5fe8f0d, 186104 bytes, 1082 packets
    out cc1346d0, 280908 bytes, 1633 packets
    local  11.11.11.11/32
    remote 22.22.22.22/32
dmvpn: #1, ESTABLISHED, IKEv2, 82fd942f9fdc4325_i 8901f24b124cbe9c_r
  local  'spoke1' @ 11.11.11.11[500]
  remote 'hub' @ 44.44.44.44[500]
  aes256-cbc/hmac-sha512/hmac-sha512/ecp384
  established 714s ago, rekeying in 9499s
  dmvpn-gre: #1, reqid 1, INSTALLED, TRANSPORT, esp:aes256-cbc/hmac-sha512
    installed 714s ago, rekeying in 5140s, expires in 5886s
    in  c481a614, 116076 bytes, 677 packets
    out cb8a052d, 21032 bytes, 126 packets
    local  11.11.11.11/32
    remote 44.44.44.44/32
```

## OSPF

### OSPF v2 Overview

OSPF is the most known routing protocol among the family of so called Link State routing protocols. The OSPF algorithm is based on the Dijkstra algorithm.

OSPF was developed by the IETF in 1988. It is described in **RFC 2328** (<https://tools.ietf.org/html/rfc2328.html>). OSPF v2 was designed as an IGP which addresses issues like scalability and convergence.

To understand OSPF advantages, it is common to compare it to the RIP routing protocol (which is a distance vector routing protocol). Compared to RIP, OSPF has the following advantages:

- OSPF is scalable, there is no hop count limitation, while RIP is limited to 15,
- As a link state protocol, OSPF converges very rapidly in comparison to RIP (which is a Distance Vector protocol),
- OSPF introduces the notion of PATH cost, while RIP only considers the cost in term of hop count,
- OSPF networks can be large and complex. This is possible thanks to the concept of OSPF areas. RIP doesn't offer this facility.

- *OSPF terminology*
- *OSPF operation*
- *OSPF packets*
- *RFC*

### OSPF terminology

It is important to understand the OSPF terminology. In this paragraph we will give the most important concepts.

**Link** An interface or router,

**Link state** The status of the link,

**Link state database** [LSD (Link State Database)] It gathers all LSA (Link State Advertisement) entries. This database is common for a defined area.

**Cost** The cost of the link, which mainly depends on the speed of interfaces. Cost is associated to interfaces, or paths.

**Area** Collection of networks or routers that have the same area identifier. 0 value is reserved for backbone operations.

---

**Note:** Within an area, each router has the same link-state information.

---

**Backbone area** In a multi-area environment, it is the transit area to which all other areas are connected (area 0)

**Stub area** Routers in this area accept routing information only from OSPF routers

**Internal router** Router having all its interfaces in a single area.

**Backbone router** Router having at least one interface in the backbone area.

**BDR (Backup Designated Router)** Designated router backup (Backup)

**DR (Designated Router)** Designated router (DR Other) Router designated by the others to represent a network. The election takes place generally by taking the lowest OSPF router-id. The election can be modified by configuring the priority of OSPF.

**ASBR** Autonomous system border router is defined by some routing information that is external to the OSPF domain.

## OSPF operation

The OSPF operation for a defined area is based on the Dijkstra algorithm. The detailed description of this mechanism in a single area or in multiple areas is out of the scope of this document.

OSPF runs directly over IP and uses protocol number 89.

## OSPF packets

OSPF exchanges information through various kinds of messages. First of all, OSPF sends type 1 hello messages to 224.0.0.5 broadcast IP address. The hello message contains information about DR and BDR. Hello message has specific fields that designates the master router and the designated backup router. Those fields are filled in by exchanging those hello messages. Note that the 224.0.0.5 broadcast address it not the only one to be used. Specific broadcast information to DR and BDR is using 224.0.0.6.

OSPF is a connection oriented protocol. once hello messages have been exchanged, OSPF exchanges unicast packets. Various message types can then be exchanged:

- type 2 database description. It describes the link-state database of OSPF devices. This information is sent by OSPF routing device itself.
- type 3 link state requests (LSA. It is a request from one OSPF device that needs a specific link-state database information of a remote peer.
- type 4 link state update. It is the information about link state advertisements. The LSA provides information to reach the ASBR.
- type 5 link state ack. This message acknowledges thanks to a sequence number the previous reception of a link state update.

There are subtypes of link state updates. As remind, OSPF is used to share link state database, based on the local and remote devices interfaces ( including IP, neighbors ..). On most cases, following link state updates can be found:

- type 1 router link entry

- type 2 network link entry generated by DR

Those above types can be found in configurations where backbone area is used.

If more areas are configured, a type 3 message named Summary LSA can be sent by ABR (Area Border Router). Type 4 message ( gives summary LSA information to reach the ASBR) and type 5 message (external LSA) are used on some specific cases ( this can be routes imported from other protocols like static routes, but also RIP or BGP).

## RFC

**RFC 1587** (<https://tools.ietf.org/html/rfc1587.html>): The OSPF NSSA (Not So Stubby Area) option

**RFC 2328** (<https://tools.ietf.org/html/rfc2328.html>): OSPF version 2

**RFC 5709** (<https://tools.ietf.org/html/rfc5709.html>): OSPF version 2 HMAC-SHA Cryptographic Authentication

### See also:

The *command reference* for details.

## Configuring OSPF

- *Basic elements for configuration*
- *Verifying OSPF configuration*
- *OSPF configuration in single area example*
- *OSPF configuration with BGP redistribution*
- *OSPF per interface configuration*

## Basic elements for configuration

1. Enable OSPF:

```
vrf main
  routing ospf
    router-id 10.125.0.1
    network 10.125.0.0/30 area 0
  ..
  ..
```

Above example shows an OSPF instance configured on main VRF. The configuration of the router-id is not mandatory, since an election process takes place inside OSPF: the router-id is first based on the IP given by manual

configuration, followed by the highest IP available on loopback interface, then followed by the highest IP available on non loopback interface.

Network command permits enabling OSPF on the network interface whose address and network mask is included into this prefix, and will announce a link connected to a stub or transit network defined by the interface address and prefix. As the 10.100.0.0/24 network belongs to eth1 interface, then OSPF will establish adjacencies over that interface. The network entry passed as parameter will be passed in the Type 3 LSA.

The area identifier is a 32 bit id by which an area is identified. Note that area value is 0, usually reserved for backbone operations. Having multiple areas in a complex IGP topology permits simplifying the route calculation of OSPF. Only ABR routers will know both areas defined.

It is possible to use an alternative OSPF configuration by defining networks based on interface and area configurations. Below configuration relies on interface eth0\_0 where the 10.125.0.0/30 network is configured. The below configuration assumes that there is only one IPv4 address under eth0\_0 so that both configurations are the same.

```
vrf main
  routing
    ospf
      router-id 10.125.0.1
      ..
    interface eth0_0
      ip ospf area 0
      ..
    ..
  interface physical eth0_0
    ipv4 address 10.125.0.1/30
```

Using above configuration can simplify network deployments, since the address configuration is tightly linked with the IP provisioning done on the interfaces.

You can also disable OSPF, without having to remove the configuration, by using following command:

```
vrf main
  routing ospf
    enabled false
```

Nonetheless, it is always possible to suppress OSPF configuration:

```
vrf main
  routing ospf
    del network 10.125.0.0/24
    ..
  ..
  del routing ospf
  ..
```

## Verifying OSPF configuration

The following commands can be used to verify OSPF operation.

- Display the global OSPF parameters (timers, area, router-id, etc.):

```
vrouter> show ospf
OSPF Routing Process, Router ID: 10.125.0.1
Supports only single TOS (TOS0) routes
This implementation conforms to RFC2328
RFC1583Compatibility flag is disabled
OpaqueCapability flag is disabled
Initial SPF scheduling delay 0 millise(s)
Minimum hold time between consecutive SPFs 50 millise(s)
Maximum hold time between consecutive SPFs 5000 millise(s)
Hold time multiplier is currently 1
SPF algorithm last executed 55.670s ago
Last SPF duration 62 usecs
SPF timer is inactive
LSA minimum interval 5000 msec
LSA minimum arrival 1000 msec
Write Multiplier set to 20
Refresh timer 10 sec
Number of external LSA 0. Checksum Sum 0x00000000
Number of opaque AS LSA 0. Checksum Sum 0x00000000
Number of areas attached to this router: 1
Area ID: 0.0.0.0 (Backbone)
  Number of interfaces in this area: Total: 1, Active: 1
  Number of fully adjacent neighbors in this area: 1
  Area has no authentication
  SPF algorithm executed 4 times
  Number of LSA 3
  Number of router LSA 2. Checksum Sum 0x0001701c
  Number of network LSA 1. Checksum Sum 0x00005dd4
  Number of summary LSA 0. Checksum Sum 0x00000000
  Number of ASBR summary LSA 0. Checksum Sum 0x00000000
  Number of NSSA LSA 0. Checksum Sum 0x00000000
  Number of opaque link LSA 0. Checksum Sum 0x00000000
  Number of opaque area LSA 0. Checksum Sum 0x00000000
```

- Display the OSPF v2 RIB:

```
vrouter> show ospf route
===== OSPF network routing table =====
N    10.125.0.0/24          [100] area: 0.0.0.0
                                     directly attached to eth1
```

(continues on next page)

(continued from previous page)

```

===== OSPF router routing table =====
===== OSPF external routing table =====

```

- Display the OSPF configuration for the specified interface:

```

vrouter> show ospf interface eth2
eth2 is up
  ifindex 4, MTU 1500 bytes, BW 1000 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.125.0.1/24, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 10.125.0.1, Network Type BROADCAST, Cost: 100
  Transmit Delay is 1 sec, State DR, Priority 1
  No backup designated router on this network
  Saved Network-LSA sequence number 0x80000002
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 5.118s
  Neighbor Count is 1, Adjacent neighbor count is 1

```

- Display the state of the relations with the neighbors:

```

vrouter> show ospf neighbor

```

| Neighbor ID        | Pri | State   | Dead Time | Address    | Interface       |
|--------------------|-----|---------|-----------|------------|-----------------|
| ↔RXmtL RqstL DBsmL |     |         |           |            |                 |
| 10.125.0.3         | 1   | Full/DR | 30.833s   | 10.125.0.3 | eth1:10.125.0.1 |
| ↔ 0                | 0   | 0       |           |            |                 |

- Display the OSPF v2 Link-State databases and information about LSAs (Link State Advertisements)

```

vrouter> show ospf database default

```

OSPF Router with ID (10.125.0.1)

Router Link States (Area 0.0.0.0)

| Link ID    | ADV Router | Age  | Seq#       | CkSum  | Link count |
|------------|------------|------|------------|--------|------------|
| 10.125.0.1 | 10.125.0.1 | 1171 | 0x80000007 | 0xb213 | 1          |
| 10.125.0.3 | 10.125.0.3 | 1134 | 0x80000007 | 0xae11 | 1          |

Net Link States (Area 0.0.0.0)

(continues on next page)

(continued from previous page)

| Link ID    | ADV Router | Age  | Seq#        | CkSum  |
|------------|------------|------|-------------|--------|
| 10.125.0.3 | 10.125.0.3 | 1174 | 0x800000004 | 0x57d7 |

### OSPF configuration in single area example

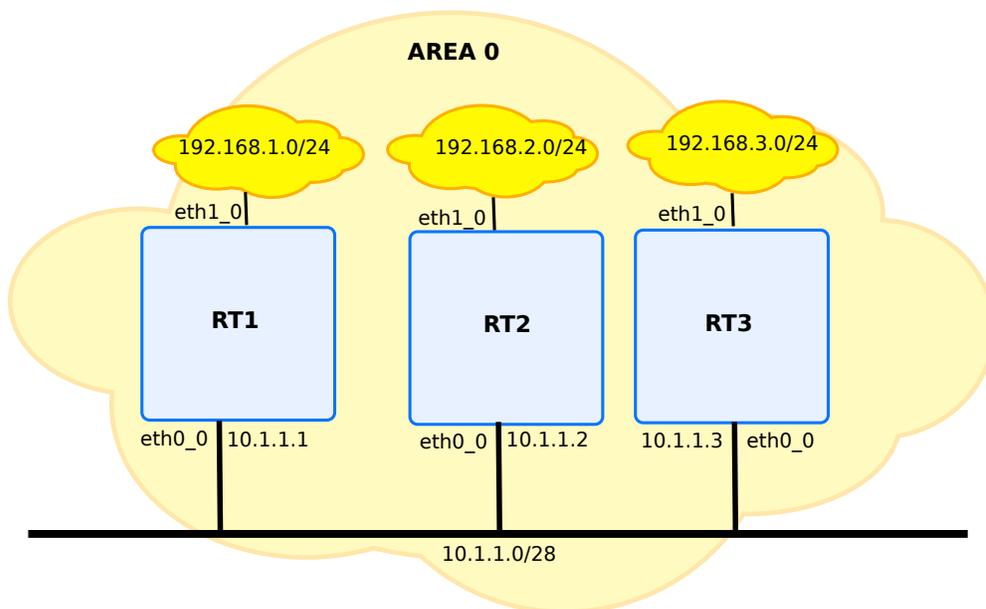


Fig. 15: First OSPF v2 configuration

rt1

```
vrf main
  routing ospf
    network 10.1.1.0/28 area 0
    ..
    ..
  interface
    physical eth1_0
      ipv4 address 192.168.1.0/24
      ..
    physical eth0_0
      ipv4 address 10.1.1.1/28
      ..
    ..
```

**rt2**

```
vrf main
  routing ospf
    network 10.1.2.0/28 area 0
    ..
    ..
  interface
    physical eth1_0
      ipv4 address 192.168.2.0/24
      ..
    physical eth0_0
      ipv4 address 10.1.1.2/28
      ..
    ..
```

**rt3**

```
vrf main
  routing ospf
    network 10.1.3.0/28 area 0
    ..
    ..
  interface
    physical eth1_0
      ipv4 address 192.168.3.0/24
      ..
    physical eth0_0
      ipv4 address 10.1.1.3/28
      ..
    ..
```

The verification of the operation can be done with following command:

```
rt1> show ospf neighbor
Neighbor ID      Pri State           Dead Time Address      Interface
↔RXmtL RqstL DBsmL
192.168.2.0      1 Full/Backup     33.553s 10.1.1.2     eth0_0:10.1.1.1
↔ 0             0 0
192.168.3.0      1 Full/DROther   37.951s 10.1.1.3     eth0_0:10.1.1.1
↔ 0             0 0
```

**Note:** The state must be Full. In this state, routers are fully adjacent with each other. All the router and network

LSAs (Link State Advertisements) are exchanged and the routers' databases are fully synchronized.

When you get used with the semantic of the OSPF v2 database, it can be displayed with the following command. The details about these entries are out of the scope of this document.

```
rt1> show ospf database default
```

```
OSPF Router with ID (192.168.2.0)
```

```
Router Link States (Area 0.0.0.0)
```

| Link ID     | ADV Router  | Age | Seq#       | CkSum  | Link count |
|-------------|-------------|-----|------------|--------|------------|
| 192.168.1.0 | 192.168.1.0 | 214 | 0x80000004 | 0xeb12 | 1          |
| 192.168.2.0 | 192.168.2.0 | 213 | 0x80000004 | 0xe713 | 1          |
| 192.168.3.0 | 192.168.3.0 | 214 | 0x80000004 | 0xe314 | 1          |

```
Net Link States (Area 0.0.0.0)
```

| Link ID  | ADV Router  | Age | Seq#       | CkSum  |
|----------|-------------|-----|------------|--------|
| 10.1.1.1 | 192.168.1.0 | 214 | 0x80000002 | 0x5d4d |

## OSPF configuration with BGP redistribution

Following example illustrates how OSPF can be used to redistribute routes to BGP. The above drawing is reused, with some changes on rt1 and rt2.

rt1

```
vrf main
  routing bgp
    router-id 10.1.1.1
    address-family
      ipv4-unicast
        redistribute ospf
        ..
      ..
    as 55
      neighbor 192.168.1.10
        remote-as 55
      ..
    ..
  ..
```

(continues on next page)

(continued from previous page)

```

routing ospf
  network 10.100.0.0/24 area 0
  router-id 10.175.0.1
  ..
  ..
interface
  physical eth1_0
  ipv4 address 192.168.1.0/24
  ..
  physical eth0_0
  ipv4 address 10.1.1.1/28
  ..
  ..

```

**rt2**

```

vrf main
  routing ospf
  network 10.1.2.0/28 area 0
  network 192.168.2.0/28 area 0
  ..
  ..
interface
  physical eth1_0
  ipv4 address 192.168.2.0/24
  ..
  physical eth0_0
  ipv4 address 10.1.1.2/28
  ..
  ..

```

The BGP routing table of rt1 is updated with the information from rt2.

```

rt1> show bgp ipv4 unicast
BGP table version is 4, local router ID is 10.1.1.1, vrf id 0
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
                i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 192.168.2.0      10.1.1.2          200           32768 ?

```

## OSPF per interface configuration

There is a wide variety of per-interface OSPF configuration items. Using same parameters between 2 OSPF instances is mandatory, and it is often useful to rely on that. Below example shows it is possible to change the network type of an interface. The OSPF network of interface `eth0_0` is defined as a non broadcast type. Adding to that configuration, the retransmit interval timer has been changed.

```
vrf main
  routing ospf
    router-id 10.125.0.1
    ..
    ..
  routing interface eth1_0
    ip ospf network non-broadcast
    ip ospf area 0
    ip ospf retransmit-interval 6
    ..
    ..
```

## Configuring OSPF in multiple areas

The need for using multiple areas is dictated by scalability issues. A single area OSPF network with many routers implies frequent SPF (Shortest Path First) calculations, large routing tables, large link-state tables, and so on...

The design of the OSPF protocol is hierarchical, that is why OSPF scales well. OSPF v2 achieves this through the use of many areas.

## OSPF operation across multiple areas

In an OSPF v2 multiple area environment the route to a specified destination is calculated as follows:

- If the destination is in the same area, the normal SPF calculation is performed
- If the destination is a network in another area, the route to the destination will be the route to the best ABR. Thus, packets addressed to the network will be received by an ABR, which will route them through the backbone area up to an ABR of the remote area. Finally, the remote ABR will forward the packets within the remote area up to the destination.

### Configuration procedure

Below drawing illustrates how to configure a backbone network with 2 devices. At each side of the 2 devices, other area are defined. As you can see, all areas have one direct link connection to area 0.

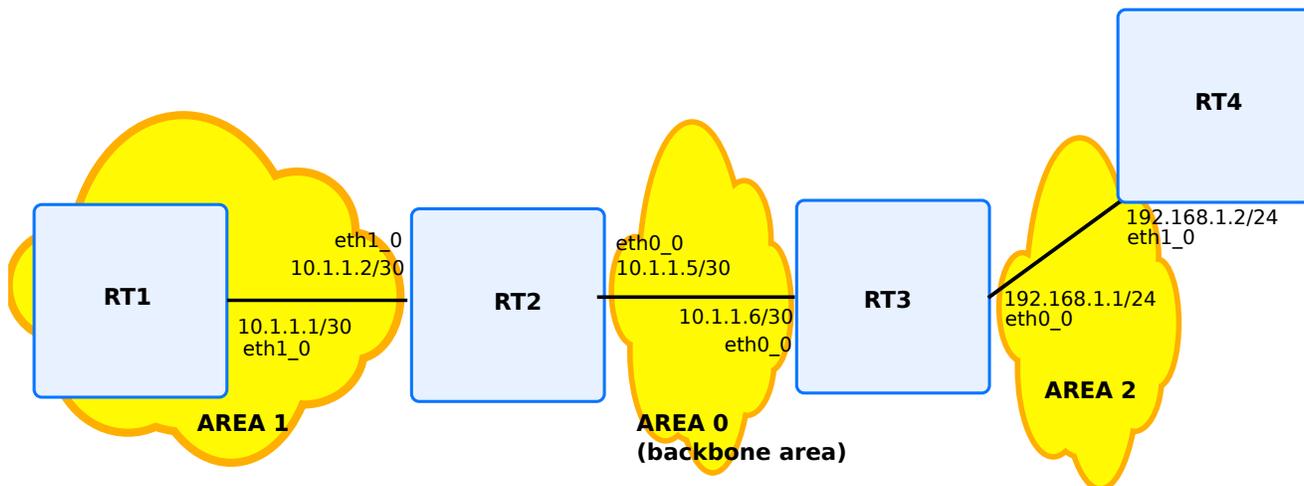


Fig. 16: OSPF v2 router configuration in multi-area environment

rt1

```
vrf main
  routing ospf
    network 10.1.1.0/30 area 1
    network 172.16.1.0/24 area 1
    ..
    ..
  interface
    physical eth0_0
      ipv4 address 172.16.1.1/24
      ..
    physical eth1_0
      ipv4 address 10.1.1.1/30
      ..
    ..
```

**rt2 (ABR between the areas 1 and 0)**

```
vrf main
  routing ospf
    network 10.1.1.0/30 area 1
    network 172.16.1.4/30 area 0
    ..
    ..
  interface
    physical eth0_0
      ipv4 address 10.1.1.5/30
      ..
    physical eth1_0
      ipv4 address 10.1.1.2/30
      ..
    ..
```

**rt3 (ABR between the areas 0 and 2)**

```
vrf main
  routing ospf
    network 10.1.1.4/30 area 0
    network 192.168.1.0/24 area 2
    ..
    ..
  interface
    physical eth0_0
      ipv4 address 10.1.1.6/30
      ..
    physical eth1_0
      ipv4 address 192.168.1.1/24
      ..
    ..
```

**rt4**

```
vrf main
  routing ospf
    network 192.168.1.0/24 area 2
    ..
    ..
  interface
```

(continues on next page)

(continued from previous page)

```

physical eth1_0
  ipv4 address 192.168.1.2/24
  ..
  ..

```

## Verifying OSPF multi-area operation

In this type of configuration, the most important thing to check is the OSPF v2 database.

### Area 1 ABR

```
rt2> show ospf database default
```

```
OSPF Router with ID (10.1.1.5)
```

```
Router Link States (Area 0.0.0.0)
```

| Link State ID | ADV Router  | Age | Seq#       | CkSum  | Link count |
|---------------|-------------|-----|------------|--------|------------|
| 10.1.1.5      | 10.1.1.5    | 53  | 0x80000004 | 0x7d84 | 1          |
| 192.168.1.1   | 192.168.1.1 | 53  | 0x80000004 | 0xfe4d | 1          |

```
Net Link States (Area 0.0.0.0)
```

| Link State ID | ADV Router  | Age | Seq#       | CkSum  |
|---------------|-------------|-----|------------|--------|
| 10.1.1.6      | 192.168.1.1 | 54  | 0x80000001 | 0x550e |

```
Summary Link States (Area 0.0.0.0)
```

| Link State ID | ADV Router  | Age | Seq#       | CkSum  | Route          |
|---------------|-------------|-----|------------|--------|----------------|
| 10.1.1.0      | 10.1.1.5    | 62  | 0x80000001 | 0x9c9c | 10.1.1.0/30    |
| 172.16.1.0    | 10.1.1.5    | 62  | 0x80000001 | 0x1c5e | 172.16.1.0/24  |
| 192.168.1.0   | 192.168.1.1 | 75  | 0x80000001 | 0xf983 | 192.168.1.0/24 |

```
Router Link States (Area 0.0.0.1)
```

| Link State ID | ADV Router | Age | Seq#       | CkSum  | Link count |
|---------------|------------|-----|------------|--------|------------|
| 10.1.1.5      | 10.1.1.5   | 62  | 0x80000003 | 0x21e8 | 1          |
| 172.16.1.1    | 172.16.1.1 | 75  | 0x80000003 | 0xeceb | 2          |

```
Net Link States (Area 0.0.0.1)
```

| Link State ID | ADV Router | Age | Seq#       | CkSum  |
|---------------|------------|-----|------------|--------|
| 10.1.1.1      | 172.16.1.1 | 77  | 0x80000001 | 0x467b |

(continues on next page)

(continued from previous page)

```

                Summary Link States (Area 0.0.0.1)
Link State ID ADV Router Age Seq#          CkSum  Route
10.1.1.4        10.1.1.5    53 0x80000001  0x74c0 10.1.1.4/30
192.168.1.0    10.1.1.5    43 0x80000001  0xefdd 192.168.1.0/24

```

rt2 has two databases: one in area 1, the other in area 0.

**rt1**

```

rt1> show ospf database default

        OSPF Router with ID (172.16.1.1)

                Router Link States (Area 0.0.0.1)

Link State ID ADV Router Age Seq#          CkSum  Link count
10.1.1.5        10.1.1.5    100 0x8000000a  0x1fe9 1
172.16.1.1     172.16.1.1  200 0x8000000b  0xeaec 2

                Net Link States (Area 0.0.0.1)

Link State ID ADV Router Age Seq#          CkSum
10.1.1.1        172.16.1.1  200 0x80000005  0x447c

                Summary Link States (Area 0.0.0.1)

Link State ID ADV Router Age Seq#          CkSum  Route
10.1.1.4        10.1.1.5    96 0x80000002  0x72c1 10.1.1.4/30
192.168.1.0    10.1.1.5    93 0x80000001  0xefdd 192.168.1.0/24

```

## Route summarization

Summarization is the aggregation of multiple routes into one advertisement. The functionality of route summarization has the obvious advantage of reducing routing tables, and positively affects the amount of bandwidth and CPU consumed, but proper summarization operation requires a contiguous network address space.

There are two types of summarization:

**Inter-area route summarization** Done on ABR routers.

**External route summarization** Done on ASBR routers, this type of summarization is specific to external routes redistributed from BGP, static, or other external routing information.

## Inter-area Route summarization configuration

### Example: inter-area route summarization configuration

Above figure 6 example (Figure 6 - OSPF v2 router configuration in multi-area environment) illustrates an inter-area configuration example. Assuming that prefix 10.2.1.0/24 has been delegated to area 1, then the area 1 administrator may want to advertise a summarized route to all sub-networks of this prefix.

In the previous example, the ABR router rt2 is now configured to advertise the aggregated prefix 10.2.1.0/24, and rt1 is configured to announce network 10.2.1.0/28.

Added configuration lines are written below:

#### rt1

```
vrf main
  routing ospf
    network 10.1.1.0/30 area 1
    network 172.16.1.0/24 area 1
    network 10.2.1.0/30 area 1
    ..
    ..
  interface
    physical eth0_0
      ipv4 address 172.16.1.1/24
      ..
    physical eth1_0
      ipv4 address 10.1.1.1/30
      ipv4 address 10.2.1.1/28
      ..
      ..
```

#### rt2

ABR between the areas 1 and 0:

```
vrf main
  routing ospf
    network 10.1.1.0/30 area 1
    network 172.16.1.4/30 area 0
    area 1 range 10.2.1.0/24
    ..
    ..
  interface
```

(continues on next page)

(continued from previous page)

```

physical eth0_0
  ipv4 address 10.1.1.5/30
  ..
physical eth1_0
  ipv4 address 10.1.1.2/30
  ..
  ..

```

Check OSPF v2 routes.

### rt1

```

rt1> show ospf route
===== OSPF network routing table =====
N   10.1.1.0/30      [100] area: 0.0.0.1
                        directly attached to eth1_0
N IA 10.1.1.4/30    [200] area: 0.0.0.1
                        via 10.1.1.2, eth1_0
N   10.2.1.0/28     [100] area: 0.0.0.1
                        directly attached to eth0_0
N   172.16.1.0/24   [100] area: 0.0.0.1
                        directly attached to eth0_0
N IA 192.168.1.0/24 [300] area: 0.0.0.1
                        via 10.1.1.2, eth1_0

===== OSPF router routing table =====
R   10.1.1.5        [100] area: 0.0.0.1, ABR
                        via 10.1.1.2, eth1_0

===== OSPF external routing table =====

```

On rt1, which is in area 1, the new route to the 10.2.1.0/28 prefix has appeared in the OSPF RIB.

### rt2

```

rt2> show ospf route
===== OSPF network routing table =====
N   10.1.1.0/30      [100] area: 0.0.0.1
                        directly attached to eth1_0
N   10.1.1.4/30      [100] area: 0.0.0.0
                        directly attached to eth0_0
D IA 10.2.1.0/24    Discard entry

```

(continues on next page)

(continued from previous page)

```

N    10.2.1.0/28          [200] area: 0.0.0.1
                                via 10.1.1.1, eth1_0
N    172.16.1.0/24       [200] area: 0.0.0.1
                                via 10.1.1.1, eth1_0
N IA 192.168.1.0/24     [200] area: 0.0.0.0
                                via 10.1.1.6, eth0_0

===== OSPF router routing table =====
R    192.168.1.1         [100] area: 0.0.0.0, ABR
                                via 10.1.1.6, eth0_0

===== OSPF external routing table =====

```

On rt2, which is the ABR of area 1, the new route to the 10.2.1.0/28 prefix has appeared in the OSPF RIB. This route will not be advertised beyond area 1. The summary route will instead be advertised. To avoid routing loops (since the 10.2.1.0/24 address space has not be entirely assigned to networks), a reject route will be injected in the ABR forwarding table (hence a discard entry appears in the OSPF RIB).

### rt3

```

rt3> show ospf route
===== OSPF network routing table =====
N IA 10.1.1.0/30         [200] area: 0.0.0.0
                                via 10.1.1.5, eth0_0
N    10.1.1.4/30         [100] area: 0.0.0.0
                                directly attached to eth0_0
N IA 10.2.1.0/24        [300] area: 0.0.0.0
                                via 10.1.1.5, eth0_0
N IA 172.16.1.0/24      [300] area: 0.0.0.0
                                via 10.1.1.5, eth0_0
N    192.168.1.0/24     [100] area: 0.0.0.2
                                directly attached to eth1_0

===== OSPF router routing table =====
R 10.1.1.5               [100] area: 0.0.0.0, ABR
                                via 10.1.1.5, eth0_0

===== OSPF external routing table =====

```

The rt3 router, does not belong to area 1. Its OSPF RIB only contains a route to the summary route 10.2.1.0/24.

### OSPF virtual links overview

When configuring OSPF in multi-area environment, one area must be defined as a backbone area, this is the area 0. All communications between two areas go through the backbone area, what means that all other areas must be directly connected to the backbone area.

In some situations, a new area is added after the OSPF network has been designed, and it is not possible to have direct connection between the backbone area and the newly added area. The concept of virtual link enables to create this direct connection.

Virtual links cannot be configured over stub area.

The virtual link has two requirements:

- It must be established between two routers in the same area
- At least one of the two routers must have a connection to the backbone area.

### Virtual links configuration example

A multi-area environment will be configured, and two routers will form the virtual link. Those two routers must be ABRs (Area Border Routers), with one router connected to the backbone area.

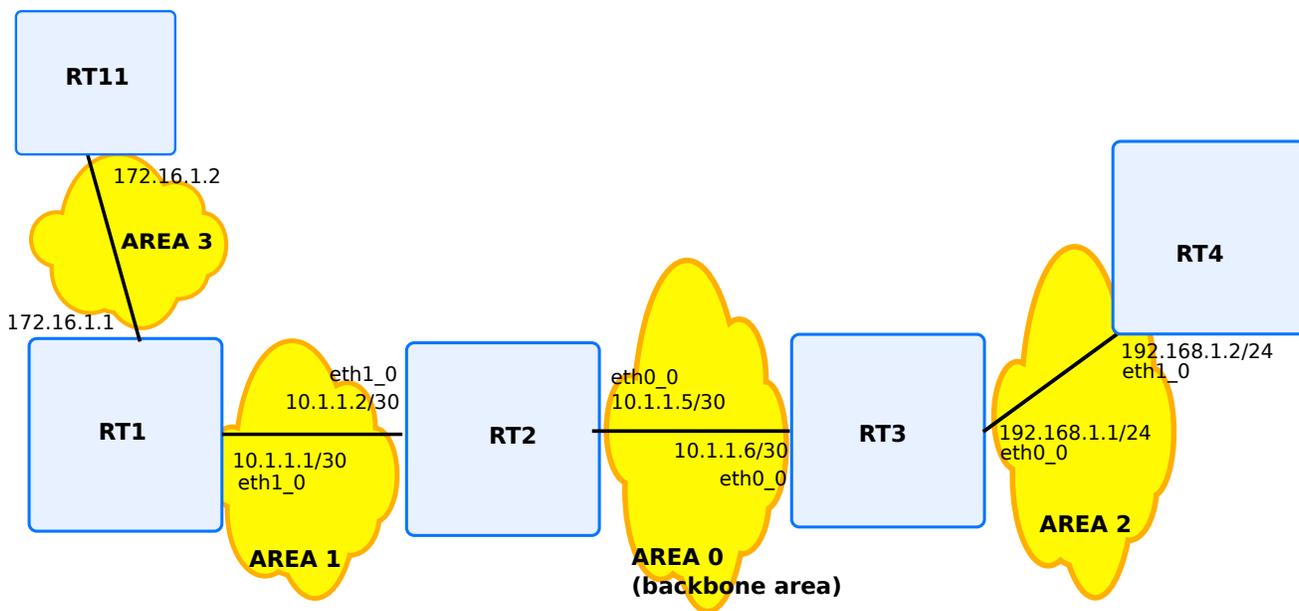


Fig. 17: OSPF v2 virtual link example

**rt11**

```
vrf main
  routing ospf
    network 172.16.1.0/24 area 3
    ..
    ..
  interface
    physical eth0_0
      ipv4 address 172.16.1.2/24
    ..
    ..
```

**rt1**

```
vrf main
  routing ospf
    area 1 virtual-link 10.1.1.5
    network 172.16.1.0/24 area 3
    network 10.1.1.0/24 area 1
    ..
    ..
  interface
    physical eth0_0
      ipv4 address 172.16.1.1/24
    ..
    physical eth1_0
      ipv4 address 10.1.1.1/30
    ..
    ..
```

**rt2**

```
vrf main
  routing ospf
    network 10.1.1.0/30 area 1
    network 10.1.1.4/30 area 0
    area 1 virtual-link 172.16.1.1
    ..
    ..
  interface
    physical eth0_0
```

(continues on next page)

(continued from previous page)

```

    ipv4 address 10.1.1.5/30
    ..
physical eth1_0
    ipv4 address 10.1.1.2/30
    ..
    ..

```

## Verifying virtual link operation

1. Check on both routers (rt1 and rt2) that the virtual link interface is up:

```

rt1> show ospf interface
[... ]
VLINK0 is up
  ifindex 0, MTU 1500 bytes, BW 0 Mbit <UP>
  Internet Address 10.1.1.1/30, Peer 10.1.1.2, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 172.16.1.1, Network Type VIRTUALLINK, Cost: 100
  Transmit Delay is 1 sec, State Point-To-Point, Priority 1
  No backup designated router on this network
  No designated router on this network
  Multicast group memberships: <None>
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 9.760s
  Neighbor Count is 1, Adjacent neighbor count is 1

```

2. Check the OSPF LSA advertisement. That is to say that rt1, which is in area 3, should receive summary link states from other areas.

```

rt11> show ospf database default

  OSPF Router with ID (172.16.1.1)

      Summary Link States (Area 0.0.0.0)

Link State ID  ADV Router  Age  Seq#      CkSum  Route
10.1.1.0        10.1.1.5    1145  0x80000001  0x9c9c  10.1.1.0/30
10.1.1.0        172.16.1.1  324   0x80000001  0x8407  10.1.1.0/30
172.16.1.0     172.16.1.1  1148  0x80000001  0x9f37  172.16.1.0/24
192.168.1.0    192.168.1.1 1142  0x80000001  0xf983  192.168.1.0/24
[... ]

      Summary Link States (Area 0.0.0.1)

```

(continues on next page)

(continued from previous page)

| Link State ID                      | ADV Router | Age  | Seq#       | CkSum  | Route          |
|------------------------------------|------------|------|------------|--------|----------------|
| 10.1.1.4                           | 10.1.1.5   | 1145 | 0x80000001 | 0x74c0 | 10.1.1.4/30    |
| 172.16.1.0                         | 172.16.1.1 | 132  | 0x80000002 | 0x9d38 | 172.16.1.0/24  |
| 192.168.1.0                        | 10.1.1.5   | 1094 | 0x80000001 | 0xefdd | 192.168.1.0/24 |
| [...]                              |            |      |            |        |                |
| Summary Link States (Area 0.0.0.3) |            |      |            |        |                |
| Link State ID                      | ADV Router | Age  | Seq#       | CkSum  | Route          |
| 10.1.1.0                           | 172.16.1.1 | 324  | 0x80000001 | 0x8407 | 10.1.1.0/30    |
| 10.1.1.4                           | 172.16.1.1 | 140  | 0x80000001 | 0xc0bc | 10.1.1.4/30    |
| 192.168.1.0                        | 172.16.1.1 | 140  | 0x80000001 | 0x3cd9 | 192.168.1.0/24 |

Moreover, this database contains the entries of the backbone area.

### OSPF stub area overview

In some ASes, the majority of the link-state database may consist of AS-external-LSAs. An OSPF AS-external-LSA is usually flooded throughout the entire AS. However, OSPF allows certain areas to be configured as “stub areas”. AS-external-LSAs are not flooded into/throughout stub areas; routing to AS external destinations in these areas is based on a default route. This reduces the link-state database size, and therefore the memory requirements, for a stub area’s internal routers.

To configure a stub area, enter for example:

```
routing ospf
  area 1 stub
```

### Totally stubby area overview

This feature prevents the ospf ABR from injecting inter-area summary into the considered area.

A Stub Area restricts the LSA types being injected into a stub area from other areas to Type 3 Summary LSA’s. Type 4’s and 5’s are represented by a default route to the Area Border Router. A totally stubby area takes this further by restricting Type 3’s as well, so all traffic being injected into a totally stubby area are represented by a default route.

To sum up, this means that the AS-external-LSAs (Type-5 LSA) and ASBR-Summary-LSA (Type-4 LSA) and Network summary LSA (Type-3 LSA) are not flooded into a totally stub areas.

### Example

```
vrf main
  routing ospf
    area 1 stub summary false
```

### OSPF NSSA overview

Turbo Router software supports the OSPF NSSA. This concept was first described in **RFC 1587** (<https://tools.ietf.org/html/rfc1587.html>). An OSPF area is said to be NSSA if it can send some external links to other areas. These routes are said to be LSA type 7, which carry essentially type 5 LSA. Then, at the ASBR, it is converted in LSA type 5, which can flood the information to the rest of other areas networks.

### Example

```
vrf main
  routing ospf
    area 1 nssa
```

### OSPF options configuration example

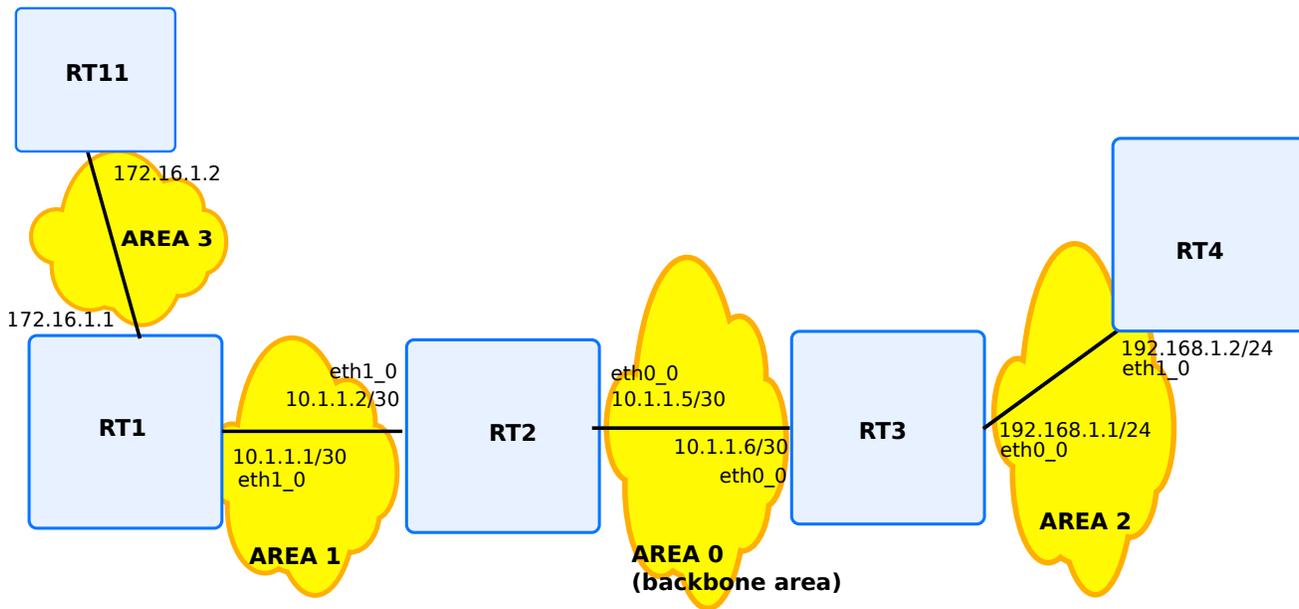


Fig. 18: OSPF v2 options configuration example

In this example, the routers will be configured so that rt1 and rt2 will have a virtual-link. Route summarization will be configured on rt1. rt2 and rt3 will be ABRs. Also, OSPF priority on rt2 will be changed. The last device, rt3, will be configured in area 2. It will be checked how routes announced by rt1 will be propagated.

### rt11

```
vrf main
  routing ospf
    network 172.16.0.0/22 area 3
    ..
    ..
  interface
    physical eth0_0
      ipv4 address 172.16.1.2/24
      ..
    physical eth1_0
      ipv4 address 172.16.0.2/24
      ..
    ..
```

### rt1

```
vrf main
  routing ospf
    area 1 virtual-link 10.1.1.5
    area 3 range 172.16.0.0/22
    network 172.16.0.0/22 area 3
    network 10.1.1.0/24 area 1
    ..
    ..
  interface
    physical eth0_0
      ipv4 address 172.16.1.1/24
      ..
    physical eth1_0
      ipv4 address 10.1.1.1/30
      ..
    ..
```

**rt2**

```
vrf main
  routing ospf
    network 10.1.1.0/30 area 1
    network 10.1.1.4/30 area 0
    area 1 virtual-link 172.16.1.1
    ..
    ..
  routing interface eth1_0
    ip ospf priority 3
    ..
    ..
  interface
    physical eth0_0
      ipv4 address 10.1.1.5/30
      ..
    physical eth1_0
      ipv4 address 10.1.1.2/30
      ..
    ..
```

**rt3**

```
vrf main
  routing ospf
    network 10.1.1.4/30 area 0
    network 192.168.1.0/24 area 2
    ..
  interface
    physical eth0_0
      ipv4 address 10.1.1.6/30
      ..
    physical eth1_0
      ipv4 address 192.168.1.1/24
      ..
    ..
```

**rt4**

```
vrf main
  routing ospf
    network 192.168.1.0/24 area 2
    ..
    ..
  interface
    physical eth1_0
    ipv4 address 192.168.1.2/24
    ..
    ..
```

Check the state of the multi-area OSPF domain.

**rt1**

Check the OSPF neighbors' status:

```
rt1> show ospf neighbor
```

| Neighbor ID | Pri | State        | Dead Time | Address    | Interface         |
|-------------|-----|--------------|-----------|------------|-------------------|
| 10.1.1.5    | 3   | Full/DR      | 35.432s   | 10.1.1.2   | eth1_0:10.1.1.1   |
| 0           | 0   | 0            |           |            |                   |
| 10.1.1.5    | 1   | Full/DROther | 34.433s   | 10.1.1.2   | VLINK0            |
| 0           | 0   | 0            |           |            |                   |
| 172.16.1.2  | 1   | Full/DR      | 31.642    | 172.16.1.2 | eth0_0:172.16.1.1 |
| 0           | 0   | 0            |           |            |                   |

**rt2**

Check the OSPF neighbors' status:

```
rt2> show ospf neighbor
```

| Neighbor ID | Pri | State        | Dead Time | Address  | Interface       | RXmtL | RqstL | DBsmL |
|-------------|-----|--------------|-----------|----------|-----------------|-------|-------|-------|
| 172.16.1.1  | 1   | Full/Backup  | 38.325s   | 10.1.1.1 | eth1_0:10.1.1.2 | 0     | 0     | 0     |
| 192.168.1.1 | 1   | Full/DR      | 38.635s   | 10.1.1.6 | eth0_0:10.1.1.5 | 0     | 0     | 0     |
| 172.16.1.1  | 1   | Full/DROther | 38.405s   | 10.1.1.1 | VLINK0          | 0     | 0     | 0     |

**rt3**

Check the OSPF neighbors' status:

```

rt3> show ospf neighbor
Neighbor ID Pri State Dead Time Address Interface RXmtL RqstL
↪DBsmL
192.168.1.2 1 Full/DR 36.257s 192.168.1.2 eth1_0:192.168.1.1 0 0
↪ 0
10.1.1.5 1 Full/Backup 32.532s 10.1.1.5 eth0_0:10.1.1.6 0 0
↪ 0

```

**rt2**

Display the OSPF database:

```

rt2> show ospf database default

OSPF Router with ID (10.1.1.5)

Router Link States (Area 0.0.0.0)

Link State ID ADV Router Age Seq# CkSum Link count
10.1.1.5 10.1.1.5 601 0x80000001b 0x827f 2
172.16.1.1 172.16.1.1 598 0x800000010 0x5844 1
192.168.1.1 192.168.1.1 649 0x800000010 0xe45a 1

Net Link States (Area 0.0.0.0)

Link State ID ADV Router Age Seq# CkSum
10.1.1.6 192.168.1.1 653 0x800000001 0x550e

Summary Link States (Area 0.0.0.0)

Link State ID ADV Router Age Seq# CkSum Route
10.1.1.0 10.1.1.5 990 0x800000005 0x94a0 10.1.1.0/30
10.1.1.0 172.16.1.1 979 0x800000005 0x7c0b 10.1.1.0/30
172.16.0.0 172.16.1.1 657 0x800000001 0x9b3f 172.16.0.0/22
192.168.1.0 192.168.1.1 626 0x800000006 0xef88 192.168.1.0/24

Router Link States (Area 0.0.0.1)

Link State ID ADV Router Age Seq# CkSum Link count
10.1.1.5 10.1.1.5 602 0x800000004 0x35ce 1
172.16.1.1 172.16.1.1 603 0x800000005 0x3e6a 1

```

(continues on next page)

(continued from previous page)

| Net Link States (Area 0.0.0.1)     |               |     |            |        |                |
|------------------------------------|---------------|-----|------------|--------|----------------|
| Link State                         | ID ADV Router | Age | Seq#       | CkSum  |                |
| 10.1.1.2                           | 10.1.1.5      | 611 | 0x80000001 | 0x541a |                |
| Summary Link States (Area 0.0.0.1) |               |     |            |        |                |
| Link State                         | ID ADV Router | Age | Seq#       | CkSum  | Route          |
| 10.1.1.4                           | 10.1.1.5      | 649 | 0x80000001 | 0x74c0 | 10.1.1.4/30    |
| 172.16.0.0                         | 172.16.1.1    | 657 | 0x80000001 | 0x9b3f | 172.16.0.0/22  |
| 172.16.0.255                       | 172.16.1.1    | 596 | 0x80000001 | 0x0fbe | 172.16.0.0/24  |
| 172.16.1.0                         | 172.16.1.1    | 596 | 0x80000001 | 0x9f37 | 172.16.1.0/24  |
| 192.168.1.0                        | 10.1.1.5      | 639 | 0x80000001 | 0xefdd | 192.168.1.0/24 |

On above show command, a summary LSA exists for networks 172.16.0.0/24 and 172.16.0.1/24 in area 1 (although these networks are in area 3), thanks to the virtual link between rt1 and rt2. The LSAs for these two networks are aggregated in area 0 as a summary link state, thanks to route summarization on router rt1, hence only a route to network 172.16.0.0/22 is advertised on the backbone area.

### rt3

Display the OSPF routes received by rt3:

```

rt3> show ospf route
===== OSPF network routing table =====
N IA 10.1.1.0/30          [200] area: 0.0.0.0
                        via 10.1.1.5, eth0_0
N   10.1.1.4/30          [100] area: 0.0.0.0
                        directly attached to eth0_0
N IA 172.16.0.0/22      [310] area: 0.0.0.0
                        via 10.1.1.5, eth0_0
N   192.168.1.0/24      [100] area: 0.0.0.2
                        directly attached to eth1_0

===== OSPF router routing table =====
R   10.1.1.5             [100] area: 0.0.0.0, ABR
                        via 10.1.1.5, eth0_0
R   172.16.1.1          [200] area: 0.0.0.0, ABR
                        via 10.1.1.5, eth0_0

===== OSPF external routing table =====

```

The aggregated route to network 172.16.0.0/22 is received by rt3 thanks to the virtual link and route summarization.

**rt1**

On rt1, the OSPF routes are as follows:

```

rt1> show ospf route
===== OSPF network routing table =====
N   10.1.1.0/30      [100] area: 0.0.0.1
                        directly attached to eth1_0
N   10.1.1.4/30      [200] area: 0.0.0.0
                        via 10.1.1.2, eth1_0
D IA 172.16.0.0/22   Discard entry
N   172.16.0.0/24    [110] area: 0.0.0.3
                        via 172.16.1.2, eth0_0
N   172.16.1.0/24    [100] area: 0.0.0.3
                        directly attached to eth0_0
N IA 192.168.1.0/24 [300] area: 0.0.0.0
                        via 10.1.1.2, eth1_0

===== OSPF router routing table =====
R   10.1.1.5         [100] area: 0.0.0.1, ABR
                        via 10.1.1.2, eth1_0
                        [100] area: 0.0.0.0, ABR
                        via 10.1.1.2, eth1_0
R   192.168.1.1     [200] area: 0.0.0.0, ABR
                        via 10.1.1.2, eth1_0

===== OSPF external routing table =====

```

The routes to area 3 networks (172.16.0.0/24 and 172.16.1.0/24) appear in the RIB, as well as a reject route to the aggregated network (172.16.0.0/22), to avoid routing loops. Only the aggregated route will be advertised to other areas. Routes to networks in remote areas have also been received by rt1.

Routes are now installed on all routers, so that packets can flow from rt11 to rt4.

**OSPF v2 security**

Security problems could lead to DoS (Denial of Service) if falsified routing information are exchanged between routers.

Turbo Router OSPF v2 implementation supports two kinds of authentication, plain text authentication and more secure MD5 authentication.

---

**Note:** If this option is adopted, then it must be configured in the whole area. For plain text authentication, passwords must be the same between neighbors.

---

## OSPF authentication configuration

### Configuring plain text authentication

1. For each interface, type the following command at the interface level:

```
vrf main
  routing interface eth0_0
    ip ospf authentication simple
    ip ospf authentication-key secret
    ..
    ..
```

The secret password is being used in the OSPF header of OSPF messages, and is in clear form.

1. Enable ospf authentication in the corresponding area, in the router ospf context.

```
vrf main
  routing ospf
    area 0 authentication
    ..
    ..
```

1. Remove the authentication password:

```
vrf main
  routing interface eth0_0
    del ip ospf authentication-key
    del ip ospf authentication
    ..
    ..
  routing ospf
    del area 0 authentication
    ..
    ..
```

### Configuring MD5 authentication

1. For each interface, type the following command at the interface level:

```
vrf main
  routing interface eth0_0
    ip ospf authentication message-digest
    ip ospf message-digest-key 1 md5 d215
```

(continues on next page)

(continued from previous page)

```
..
..
```

A key identifier is carried in OSPF messages, along with authentication crypted data, and area identifier ( by default backbone).

1. Enable context authentication in the corresponding area, in the router ospf context.

```
vrf main
  routing ospf
    area 0 authentication message-digest true
```

1. Remove the OSPF authentication and MD5 authentication secret:

```
vrf main
  routing interface eth0_0
    del ip ospf authentication
    del ip ospf message-digest-key 1
    ..
    ..
  routing ospf
    del area 0 authentication
    ..
    ..
```

## Filtering OSPF

Like for BGP protocol, it is possible to apply filtering thanks to *route map*. Below example illustrates what can be done by using *Prefix List*. OSPF will be configured to redistribute BGP entries, however some filtering will be applied.

1. Specify the prefix-list and route-map:

```
vrf main
  routing
    ipv4-prefix-list plist
      seq 1 address 10.100.0.0/24 policy permit
      seq 2 address 10.200.0.0/24 policy deny
      seq 3 address 10.150.0.0/24 policy permit
      ..
    route-map rmap seq 1 plicy permit
    route-map rmap seq 1 match ip address prefix-list plist
    ..
```

1. Configuration of a BGP instance that peers with remote located outside of OSPF area.

```
vrf main
  routing bgp
    as 55
    router-id 1.1.1.1
    neighbor 10.110.0.10 remote-as 55
    ..
    ..
```

Subsequently, some BGP routing entries will be learnt from remote.

```
rt1> show bgp ipv4 unicast
BGP table version is 9, local router ID is 1.1.1.1, vrf id 0
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete
```

| Network          | Next Hop    | Metric | LocPrf | Weight | Path |
|------------------|-------------|--------|--------|--------|------|
| *>i10.100.0.0/24 | 10.110.0.10 | 0      | 100    | 0      | i    |
| *>i10.150.0.0/24 | 10.110.0.10 | 0      | 100    | 0      | i    |
| *>i10.200.0.0/24 | 10.110.0.10 | 0      | 100    | 0      | i    |

Displayed 3 routes and 3 total paths

#### 1. Configure the route redistribution with the route-map filtering:

```
vrf main
  routing ospf
    redistribute bgp route-map rmap
```

Subsequently, the rt1 device has imported filtered BGP route entries.

```
rt1> show ospf database default

OSPF Router with ID (1.1.1.1)

      Router Link States (Area 0.0.0.0)

Link ID          ADV Router      Age  Seq#           CkSum  Link count
1.1.1.1          1.1.1.1         127 0x800000004 0xbf9a 1

      AS External Link States

Link ID          ADV Router      Age  Seq#           CkSum  Route
10.100.0.0       1.1.1.1         630 0x800000001 0xc2ff E2 10.100.0.0/24 [0x0]
```

(continues on next page)

(continued from previous page)

|            |         |  |
|------------|---------|--|
| 10.150.0.0 | 1.1.1.1 | 621 0x80000001 0x6828 E2 10.150.0.0/24 [0x0] |
|------------|---------|--|

## BFD In OSPF

With BFD usage in OSPF, the failover mechanism is greatly improved by detecting the loss of remote OSPF neighbors. Instead of relying on standard hello mechanisms, BFD permits faster convergence. To get more information on BFD, please see *BFD*.

## BFD Configuration And Monitoring In OSPF

A BFD peer session context is created, along with discovering OSPF neighbors. Due to the nature of OSPF, all created BFD peer contexts are single-hop.

```
vrf customer1
  routing ospf
    router-id 10.125.0.1
    .. ..
  routing interface eth1_0
    ip ospf area 0.0.0.1
    ip ospf track bfd
```

Then you can continue the configuration as usual. For timer settings, the default emission and reception settings are set to 300000 microseconds, which may not be what is wished. In that case, it is possible to override default timers, by configuring general timer settings. More information is given in *Configuring general BFD settings*.

```
vrouter> show ospf vrf customer1 interface eth1_0
eth1_0 is up
  ifindex 2, MTU 1500 bytes, BW 10000 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.125.0.1/24, Broadcast 10.125.0.255, Area 0.0.0.1
  MTU mismatch detection: enabled
  Router ID 10.125.0.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Backup Designated Router (ID) 10.125.0.2, Interface Address 10.125.0.1
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 5.710s
  Neighbor Count is 1, Adjacent neighbor count is 1
  BFD: Detect Multiplier: 3, Min Rx interval: 600, Min Tx interval: 600

vrouter> show ospf vrf customer1 neighbor
Neighbor ID    Pri State          Dead Time Address          Interface
↔RXmtL RqstL DBsmL
```

(continues on next page)

(continued from previous page)

```

10.125.0.2      1 Full/Backup      38.091s 10.125.0.2      eth1_0:10.125.0.1      0
↔      0      0

vrouter> show ospf vrf customer1 database router 10.125.0.2
VRF Name: r2-cust1

    OSPF Router with ID (10.254.254.2)

                Router Link States (Area 0.0.0.1)

LS age: 70
Options: 0x2 : *|---|---|E|
LS Flags: 0x3
Flags: 0x2 : ASBR
LS Type: router-LSA
Link State ID: 10.125.0.2
Advertising Router: 10.125.0.2
LS Seq Number: 80000004
Checksum: 0xb65d
Length: 36

Number of Links: 1

Link connected to: a Transit Network
(Link ID) Designated Router address: 10.125.0.2
(Link Data) Router Interface address: 10.125.0.2
Number of TOS metrics: 0
TOS 0 Metric: 10

LS age: 70
Options: 0x2 : *|---|---|E|
LS Flags: 0x6
Flags: 0x2 : ASBR
LS Type: router-LSA
Link State ID: 10.125.0.2
Advertising Router: 10.125.0.2
LS Seq Number: 80000003
Checksum: 0x9a79
Length: 36

Number of Links: 1

```

(continues on next page)

(continued from previous page)

```
Link connected to: a Transit Network
(Link ID) Designated Router address: 10.0.3.1
(Link Data) Router Interface address: 10.0.3.1
Number of TOS metrics: 0
TOS 0 Metric: 10
```

```
vrouter> show bfd vrf customer1 session single-hop destination 10.125.0.2
```

```
BFD Peer:
peer 10.125.0.2 interface eth1_0
  ID: 322201613
  Remote ID: 2746639856
  Status: up
  Uptime: 9 minute(s), 49 second(s)
  Diagnostics: ok
  Remote diagnostics: ok
  Local timers:
    Receive interval: 300ms
    Transmission interval: 300ms
    Echo transmission interval: 50ms
  Remote timers:
    Receive interval: 300ms
    Transmission interval: 300ms
    Echo transmission interval: 50ms
```

## RIP

### RIP Overview

RIP came up at the end of the 80's. It is a routing protocol that computes the shortest path between networks. It is based on the Bellman-Ford algorithm that distributes the computation of the shortest path among the nodes (routers). The metric of the path is related to the number of hops. Consequently it is one of the most famous distance vector protocol that is used on the IP networks and on the Internet.

The first release RIP v1, that is described by the IETF **RFC 1058** (<https://tools.ietf.org/html/rfc1058.html>), was designed for the IPv4 class oriented Internet. RIP v1 uses broadcast UDP on the well-known port 520.

Nowadays, the second release of RIP (RIP v2), which is described by the IETF **RFC 2453** (<https://tools.ietf.org/html/rfc2453.html>), fits the IPv4 CIDR (Classless InterDomain Routing) that uses VLSMs (Variable Length Subnet Masks). RIP v2 uses multicast UDP on the well-known group 224.0.0.9 and port 520. You can use it as in IGP within a small simple network.

The maximum network size that RIP can handle is 16 hops.

## RFC

**RFC 1058** (<https://tools.ietf.org/html/rfc1058.html>) Routing information protocol

**RFC 2453** (<https://tools.ietf.org/html/rfc2453.html>) RIP Version 2

**RFC 4822** (<https://tools.ietf.org/html/rfc4822.html>) RIPv2 Cryptographic Authentication

### See also:

The *command reference* for details.

## RIP Configuration

### Basic elements for configuration

Starting RIP can be done by using a very simple configuration. Example below illustrates a basic configuration setup with one network configured. Automatically, RIP will operate over all the interfaces where an IP address is defined, whose network address is included in the provided network prefix. Network addresses included in this prefix and defined on these interfaces will be advertised.

```
vrf main
  routing rip
    network 10.125.0.0/30
  ..
  ..
  commit
```

As mentioned in above config, RIP is activated, with providing network prefix. It is also possible to provide interface name. If an interface name is provided, RIP will then be activated on this interface and all IPv4 network prefixes defined on this interface will be advertised.

```
vrf main
  routing rip
    interface eth1_0
```

RIP can be stopped by using following command:

```
vrf main
  del routing rip
  commit
```

Alternatively, it is also possible to just disable RIP without having to remove the whole configuration.

```
vrf main
  routing rip enabled false
  commit
```

Currently, only one RIP instance is supported for the whole Turbo Router. However, it is possible to store the configuration and set it to false. Below example illustrates that only rip instance from VRF vrf1 is available on the Turbo Router.

```
vrf main
  routing rip enabled false
  routing rip network 1.2.3.0/24
  commit
  ..
  ..
vrf vrf1
  routing rip network 5.5.5.0/24
  commit
  ..
  ..
```

## Verifying RIP configuration

The following commands can be used to verify RIP operation.

### show rip

This command displays the RIB of the RIP protocol.

```
vrouter> show rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
(n) - normal, (s) - static, (d) - default, (r) - redistribute,
(i) - interface
```

|      | Network        | Next Hop | Metric | From     | Tag | Time  |
|------|----------------|----------|--------|----------|-----|-------|
| C(i) | 10.1.1.0/28    | 0.0.0.0  | 1      | self     | 0   |       |
| C(i) | 192.168.1.0/24 | 0.0.0.0  | 1      | self     | 0   |       |
| R(n) | 192.168.2.0/24 | 10.1.1.2 | 2      | 10.1.1.2 | 0   | 02:36 |
| R(n) | 192.168.3.0/24 | 10.1.1.3 | 2      | 10.1.1.3 | 0   | 02:29 |

The display of `show rip` is composed of 7 columns, and describes the RIB of the RIP routing protocol:

**Code** describes the RIB source, the different codes are explained in the beginning of the output of `show rip` command

**Network** describes the learnt prefix (Destination prefix) with its subnet mask

**Next Hop** indicates the next hop to this destination (0.0.0.0 means itself).

**Metric** indicates the hop count to the destination prefix

**From** indicates the router that advertises the destination prefix

**Tag** this tag normally should be set to 0

**Time** the validity time. By default, it is set to 3 minutes when a RIP route is received.

### show rip status

This command displays Turbo Router running state of RIP.

```
vrouters> show rip status
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 18 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is 1
  Default redistribution metric is 1
  Redistributing:
  Default version control: send version 2, receive any version
  Interface      Send Recv Key-chain
  eth0_0         2      1 2
  eth1_0         2      1 2
  Routing for Networks:
    10.1.1.0/28
    192.168.2.0/24
    eth1_0
  Routing Information Sources:
  Gateway        BadPackets BadRoutes Distance Last Update
  10.125.0.2     0           0         120    00:00:07
  Distance: (default is 120)
```

This command gives the following information about RIP:

- The interfaces on which RIP has subscribed to the multicast group
- RIP timers
- Access-lists configured
- Redistribution configured
- RIP version configured (version 2 is the default)
- Interfaces participating in RIP updates (or RIP multicast group).
- Routing sources
- Gateways (in this case they are the RIP neighbors)
- Administrative distance

**See also:**

See the corresponding RIP options described in this document.

**RIP configuration options**

Several options are available to tune the default RIP configuration.

**Enabling ECMP**

- Configure RIP to allow equal cost multipath:

```
vrf main
  routing rip
    allow-ecmp true
  ..
..
```

**Specifying the RIP version**

By default, RIP is configured to handle both incoming v1 and v2 requests. However, it is possible to globally configure the default RIP version. Below configuration example illustrates how to disable v1.

```
vrf main
  routing rip
    version
      receive 2
      send 2
    ..
  ..
..
```

It is also possible to disable some rip versioning handling per interface. Below example illustrates how to handle both reception and emission with RIP v1 only:

```
vrf main
  routing interface eth1_0
    ip rip version send 1
    ip rip version receive 1
  ..
  ..
..
```

Above configuration can be checked by using following show command:

```
vrouter> show rip status
[...]
Routing Protocol is "rip"
Sending updates every 30 seconds with +/-50%, next due in 18 seconds
Timeout after 180 seconds, garbage collect after 120 seconds
Outgoing update filter list for all interface is not set
Incoming update filter list for all interface is not set
Default redistribution metric is 1
Redistributing: connected
Default version control: send version 1, receive version 1
  Interface      Send  Recv  Key-chain
  eth1_0         1     1
[...]
```

---

**Note:** These commands can be useful to interconnect some old RIP v1 networks to a new RIP v2 network, or during a migration period.

---

## Passive interface

A passive RIP interface can receive and process the RIP packets, however it does not send any RIP information (except to the neighbor listed by the neighbor command).

- Make an interface passive:

```
vrf main
  routing rip
    passive-interface eth1_0
    network 10.1.1.0/28
  ..
..
```

This appears on the configuration as

```
vrouter> show config vrf main routing
routint rip
  network 10.1.1.0/28
  passive-interface eth1_0
  ..
..
```

In this example, routing updates will not be advertised out the interface eth1\_0.

## Unicast announces

Although RIP v1 is a broadcast protocol and RIP v2 is a multicast protocol, the RIP routing updates can be unicasted too. Consequently, the IPv4 address of the unicast neighbors can be defined in order for RIP to send the routing updates to a set of specific RIP nodes.

To add the address of the neighbors, use the following command :

```
vrf main
  routing rip
    neighbor 10.125.0.2
  ..
..
```

---

### Note:

- This command is not required to enable RIP on point-to-point interfaces or tunnels, the network command is enough to activate RIP on these interfaces.
  - This command does NOT prevent RIP multicast packet to be sent on an interface. To suppress any RIP multicast packets, this command must be used jointly with the passive-interface command
- 

## Modifying timers

The routing protocols are based on many timers that control the stability of your network and the time convergence of the algorithms. RIP is based on three timers:

The update-interval default value is 30 seconds. This is the time between each update message emission.

The holddown interval default value is 180 seconds.

The flush interval default value is 120 seconds.

It is possible to change the timers values by using following command:

```
vrf main
  routing rip
    timers update-interval 30 holddown-interval 180 flush-interval 120
  ..
..
```

It is possible to check the timers values by using following show command:

```
vrouter> show rip status
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 9 seconds
```

(continues on next page)

(continued from previous page)

```
Timeout after 180 seconds, garbage collect after 120 seconds  
[...]
```

---

**Note:** Do not change any default value if you are deploying a RIP network over a LAN (Local Area Network). They should only be changed over some very low bandwidth links (about 32 Kbit/s or less) or over cost expensive links.

---

## Split horizon

When split-horizon is used, the learnt prefixes are not announced on the interface from which they come from. It has been designed to decrease traffic load and to avoid routing loops. To decrease the traffic load when the routing table is advertised, split-horizon is activated by default on each interface.

- Disable split-horizon:

```
vrf main  
  routing interface eth0_0  
    ip rip split-horizon disabled  
    ..  
  ..  
  ..
```

- Enable split-horizon:

```
vrf main  
  routing interface eth0_0  
    ip rip split-horizon simple  
    ..  
  ..  
  ..
```

---

### Note:

- Split-horizon is enabled or disabled on a per interface basis, and the corresponding commands are executed at the interface level.
  - Disable split-horizon when many interfaces on a broadcast area do not share the same connected prefix. In this case, it is enough to disable split-horizon on the routers that have the common connected prefixes because it will act as a gateway for the different connected prefixes.
-

## Split horizon Example

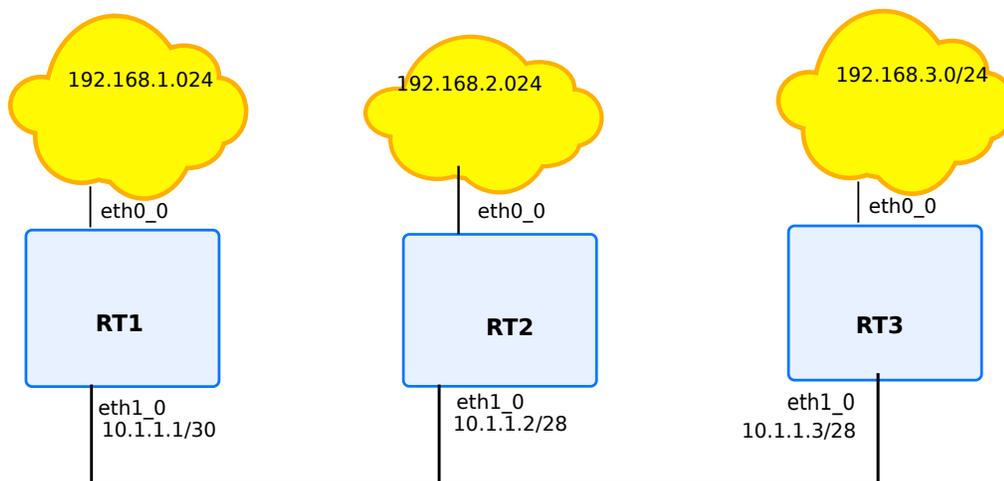


Fig. 19: RIP v2 split-horizon

To enable RIP and to demonstrate the split-horizon feature, the above figure will be used.

1. Announce the different networks:

The announcing of networks is configured like below on rt1:

```
vrf main
  routing interface eth1_0
    ip rip split-horizon simple
    ..
  ..
  routing rip
    network 10.1.1.0/28
    interface eth0_0
    ..
  ..
```

1. Show routing information:

Now RIP is running and RIP does not announce the learnt prefixes on the interfaces from which they were learnt. This is the default behavior of Turbo Router.

## Example

For example, rt1's RIP RIB is:

```
rt1> show rip
```

Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP

Sub-codes:

(n) - normal, (s) - static, (d) - default, (r) - redistribute,  
(i) - interface

|      | Network        | Next Hop | Metric | From     | Tag | Time  |
|------|----------------|----------|--------|----------|-----|-------|
| C(i) | 10.1.1.0/28    | 0.0.0.0  | 1      | self     | 0   |       |
| C(i) | 192.168.1.0/24 | 0.0.0.0  | 1      | self     | 0   |       |
| R(n) | 192.168.2.0/24 | 10.1.1.2 | 2      | 10.1.1.2 | 0   | 02:40 |
| R(n) | 192.168.3.0/24 | 10.1.1.3 | 2      | 10.1.1.3 | 0   | 02:53 |

By default, with this previous configuration, rt1 does not announce 192.168.2.0/24, neither 192.168.3.0/24 on the eth1\_0 interface due to the split-horizon feature. When split-horizon is disabled, they are announced.

## Split horizon with poisoned reverse

The goal of poisoning the reverse path is to increase the convergence of the RIP algorithm to quickly kill the RIP routing loops. When split-horizon with poisoned reverse path is enabled, the prefixes which are learned via an interface, are announced back each 30 seconds with a metric of 16 (i.e. infinite).

To increase the time convergence of the RIP algorithm, the originator routes may be poisoned. It means that the routes will be announced with an infinite metric (16) via the interface that should be used for the shortest path. However it increases the traffic load. By default Turbo Router does not activate the split-horizon with poisoned reverse path on each interface.

- Enable split-horizon with poisoned reverse path:

```
vrf main
  routing interface eth0_0
    ip rip split-horizon poisoned-reverse
  ..
..
```

- Disable the poisoned-reverse option:

```
vrf main
  routing interface eth0_0
    ip rip split-horizon simple
  ..
..
```

This will disable the poisoned-reverse option in the RIP configuration. It will fall back to the default split-horizon option.

The split horizon with poisoned reverse policy is configured on a per interface basis.

## Next-hop option

When sending a RIP message, the router will if necessary add a next-hop option to the routes it advertises. This option indicates the gateway via which the router can reach the advertised destinations. It enables the routers that receive the RIP message to create local shortcuts.

If the next-hop option is not set, then the router that originated the RIP packet is used as the next-hop.

## Default route advertisement

It is possible to force the next-hop value by using the `default-information originate` keyword.

Allow RIP to advertise the default route `0.0.0.0/0`:

```
vrf main
  routing rip
    default-information-originate true
  ..
  ..
```

- Do not advertise the default route:

```
vrf main
  routing rip
    default-information-originate false
  ..
  ..
```

---

### Note:

- When a router is advertising a default route, it is advised that it is itself configured with its own default IPv4 route to avoid that it becomes a blackhole:

```
vrf main
  routing static ipv4-route 0.0.0.0/0 next-hop 10.1.1.2
  ..
```

It is also possible to use the command `redistribute static` under `routing rip` mode, when a static route is defined.

## Using route-map to change next-hop

It is possible to configure a route-map with a a set clause. More information on route-map is given in *route map*. Below example illustrates a RIP configuration, where nexthop is forced to be a hard set value.

```
vrf main
  routing rip
    interface eth1_0
      route-map eth1_0 out route-map-name rmap_name
      ..
    ..
  ..
routing
  route-map rmap_name seq 11
  policy permit
  set ip next-hop 10.1.0.101
  ..
  ..
```

### Example

### Example

For example, if rt3 has a static route to the network 172.16.1.0/24 via a gateway - 10.1.1.4 - on the eth1\_0 interface, rt2 and rt1 know that they can directly reach this gateway without sending packets to rt3, so they conclude that there is a shorter route to network 172.16.1.0/24 via the 10.1.1.4 gateway.

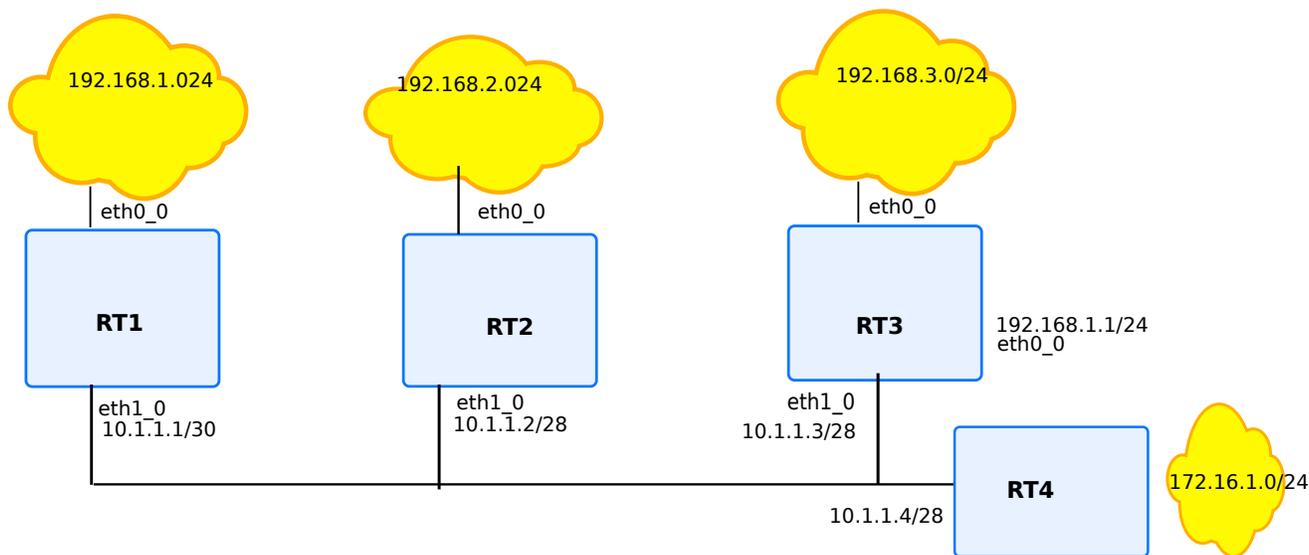


Fig. 20: Next-hop feature

Assuming the following configuration:

**rt1**

```
vrf main
  routing rip
    network 10.1.1.0/28
    network 192.168.1.0/24
  ..
..
```

**rt2**

```
vrf main
  routing rip
    network 10.1.1.0/28
    network 192.168.2.0/24
  ..
..
```

**rt3**

```
vrf main
  routing rip
    network 10.1.1.0/28
    network 192.168.3.0/24
    ..
  ..
```

It leads to the following IPv4 FIB:

```
rt3> show rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
(n) - normal, (s) - static, (d) - default, (r) - redistribute,
(i) - interface
```

|      | Network        | Next Hop | Metric | From     | Tag | Time  |
|------|----------------|----------|--------|----------|-----|-------|
| C(i) | 10.1.1.0/28    | 0.0.0.0  | 1      | self     | 0   |       |
| S(r) | 172.16.1.0/24  | 10.1.1.4 | 1      | self     | 0   |       |
| R(n) | 192.168.1.0/24 | 10.1.1.1 | 2      | 10.1.1.1 | 0   | 02:39 |
| R(n) | 192.168.2.0/24 | 10.1.1.2 | 2      | 10.1.1.2 | 0   | 02:20 |
| C(i) | 192.168.3.0/24 | 0.0.0.0  | 1      | self     | 0   |       |

While on rt2 we have:

```
rt2> show rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
(n) - normal, (s) - static, (d) - default, (r) - redistribute,
(i) - interface
```

|      | Network        | Next Hop | Metric | From     | Tag | Time  |
|------|----------------|----------|--------|----------|-----|-------|
| C(i) | 10.1.1.0/28    | 0.0.0.0  | 1      | self     | 0   |       |
| R(n) | 172.16.1.0/24  | 10.1.1.4 | 2      | 10.1.1.3 | 0   | 02:45 |
| R(n) | 192.168.1.0/24 | 10.1.1.1 | 2      | 10.1.1.1 | 0   | 02:46 |
| C(i) | 192.168.2.0/24 | 0.0.0.0  | 1      | self     | 0   |       |
| R(n) | 192.168.3.0/24 | 10.1.1.3 | 2      | 10.1.1.3 | 0   | 02:45 |

rt1 and rt2 are using the same next-hop to join the network 172.16.1.0/24 without sending the data to rt3 that originates the route.

**Note:** When the next-hop is not reachable, the router should use the originator of the RIP packet as the gateway. Then, if this originator is not reachable too, the RIP entry should be ignored. Another router could announce better

information.

---

## Static RIP route

The RIP process can announce a route that has no origin. It means that it has not been introduced into the RIP RIB by the redistribute command.

- Add a route into the RIP RIB:

```
vrf main
  routing rip
    static-route 1.2.2.0/24
  ..
..
```

---

**Note:** Configuring a static RIP route is very useful for testing purpose.

---

## Redistribute other IGPs, static routes or connected routes

The RIP signaling process can learn the network prefixes either from another routing protocol such as BGP or OSPF from the connected network prefixes that have been set on the interfaces, or from the static routes that have been set.

- Redistribute prefixes:

```
vrf main
  routing rip
    redistribute connected
    redistribute static
    redistribute bgp
    redistribute ospf
  ..
..
```

The redistribution of static routes applies to the default route too. It is a good practice to announce the default route from a CPE that provides a NAT service for the traffic through the public interface.

---

**Note:** The prefixes, which are announced with the redistribute command, are named Connected-redistribute (C(r)).

---

Redistributed connected routes appear with the sub-code C(r) in the `show rip` output.

Default route appears with the (d) sub-code, while a connected interface (announced in the router rip context with the network {A.B.C.D/M|IFNAME} command) appears with the (i) sub-code.

---

**Note:** If the same prefix is learnt via different means (redistribution, interface, or default) the route learnt via redistribution is the less preferred.

---

### FIB's RIP administrative distance

When many IGPs and EGPs (External Gateway Protocols) are provisioning a same active route into the IPv4 FIB, the one from the preferred routing protocols is selected; for example the static routes are preferred to the OSPF v2 routes that are preferred to the RIP routes that are preferred to the eBGP routes.

The default RIP distance is 120. It is however possible to override that behaviour by using the following command:

```
vrf main
  routing rip
    administrative-distance default 123
  ..
  ..
```

More information about administrative distance of other routing protocols can be found on following reference *Administrative Distance*

### Manage the redistributed metrics

Since the routing protocols are not the same (BGP, static, connected), the associated metrics cannot be compared, and hence cannot be kept within the RIP advertisements. An arbitrary distance, which is assimilated to a hop count, can be set with the redistribute SOURCE metric N command into the RIP context.

```
vrf main
  routing rip
    redistribute static metric 3
    redistribute connected metric 2
    redistribute bgp metric 9
    redistribute ospf metric 4
  ..
  ..
```

---

**Note:** Due to the maximum RIP metric (16), these commands decrease the size of your network.

---

The default redistribution metric into RIP is 1.

---

**Note:** When redistributing a routing protocol into RIP, special care must be taken for the metric control, because

not all routing protocols have the same metric. Remember that RIP uses the hop count as metric.

---

### RIP options example

In this example we will configure 4 routers rt1, rt2, rt3 and rt4 to support the RIP options.

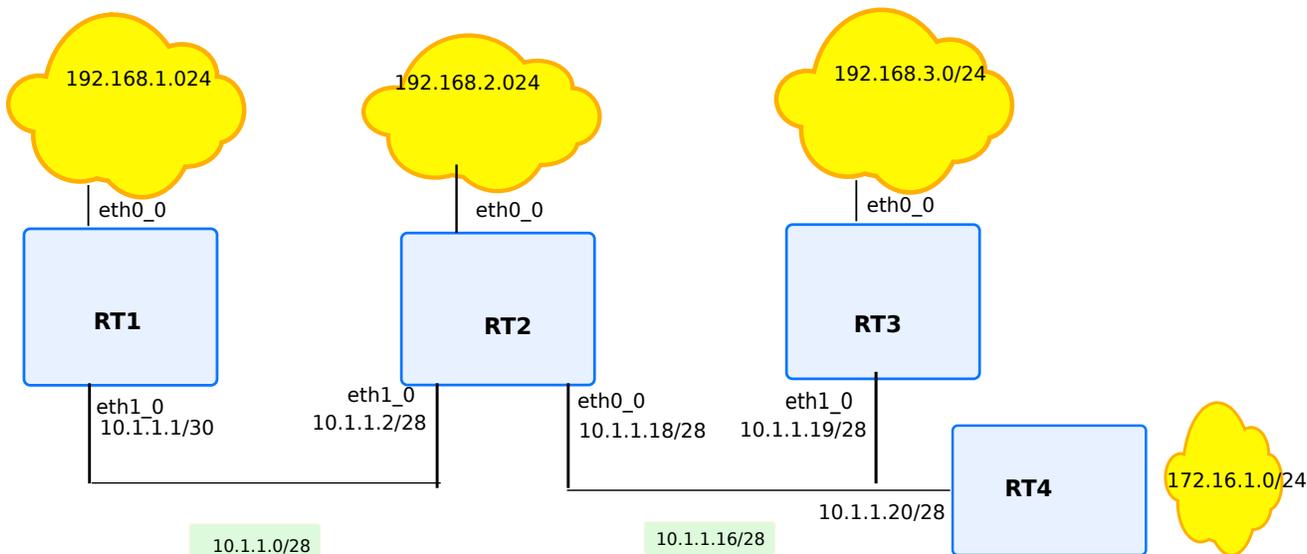


Fig. 21: RIP options

### Required features

- rt1** RIP static route option
- rt2** Delete split-horizon, poison-reverse and administrative distance options
- rt3** Redistribute connected + metric option
- rt4** Modify timers option

**rt1**

```
vrf main
  interface
    physical eth0_0
      ipv4 address 192.168.1.1/24
    ..
    physical eth1_0
      ipv4 address 10.1.1.1/28
    ..
  ..
  routing rip
    network 10.1.1.0/28
    network 192.168.1.0/24
    static-route 192.168.4.0/24
    ..
  ..
  routing static ipv4-route 192.168.4.0/24 next-hop 192.168.1.25
```

**rt2**

```
vrf main
  interface
    physical eth0_0
      ipv4 address 10.1.1.18/28
    ..
    physical eth1_0
      ipv4 address 10.1.1.2/28
    ..
    physical eth2_0
      ipv4 address 192.168.2.2/24
    ..
  ..
  routing
    interface eth0_0
      ip rip split-horizon poisoned-reverse
    ..
    interface eth1_0
      ip rip split-horizon disabled
    ..
  rip
    network 10.1.1.0/27
    network 192.168.2.0/24
```

(continues on next page)

(continued from previous page)

```
..
..
```

**rt3**

```
vrf main
interface
  physical eth0_0
    ipv4 address 192.168.3.3/24
  ..
  physical eth1_0
    ipv4 address 10.1.1.19/28
  ..
  ..
routing rip
  network 10.1.1.16/28
  redistribute connected metric 4
```

**rt4**

```
vrf main
interface
  physical eth0_0
    ipv4 address 172.16.1.4/24
  ..
  physical eth1_0
    ipv4 address 10.1.1.20/28
  ..
  ..
routing rip
  network 10.1.1.16/28
  network eth0_0
  timers update-interval 30 holddown-interval 180 flush-interval 120
  ..
  ..
```

Here is what rt1 RIP RIB and FIB look like:

```
rt1> show rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
```

(continues on next page)

(continued from previous page)

(n) - normal, (s) - static, (d) - default, (r) - redistribute,  
(i) - interface

|      | Network        | Next Hop | Metric | From     | Tag | Time  |
|------|----------------|----------|--------|----------|-----|-------|
| C(i) | 10.1.1.0/28    | 0.0.0.0  | 1      | self     | 0   |       |
| R(n) | 10.1.1.16/28   | 10.1.1.2 | 2      | 10.1.1.2 | 0   | 02:53 |
| R(n) | 172.16.1.0/24  | 10.1.1.2 | 3      | 10.1.1.2 | 0   | 02:53 |
| C(i) | 192.168.1.0/24 | 0.0.0.0  | 1      | self     | 0   |       |
| R(n) | 192.168.2.0/24 | 10.1.1.2 | 2      | 10.1.1.2 | 0   | 02:53 |
| R(n) | 192.168.3.0/24 | 10.1.1.2 | 6      | 10.1.1.2 | 0   | 02:53 |
| R(s) | 192.168.4.0/24 | 0.0.0.0  | 1      | self     | 0   |       |

The 10.1.1.0/28 and 192.168.1.0/24 routes are routes to directly connected interfaces (C(i) flag), their next hop is consequently rt1 itself and the metric is 1. The 192.168.4.0/24 route is redistributed from a static route (R(s) flag), its next hop is consequently rt1 itself and the metric is 1.

The 10.1.1.16/28, 172.16.1.0/24, and 192.168.2.0/24 routes were acquired via the RIP protocol (R(n) flag), their next hop is rt2 and their metrics correspond to the number of hops up to the destination. The 192.168.3.0/24 route's metric is 6 instead of 2, due to rt3 configuration, which increased the metric by 4.

```
rt1> show ipv4-routes
```

```
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR,
       > - selected route, * - FIB route
```

```
C>* 10.1.1.0/28 is directly connected, eth1_0
R>* 10.1.1.16/28 [120/2] via 10.1.1.2, eth1_0, 00:00:16
C>* 127.0.0.0/8 is directly connected, lo0
R>* 172.16.1.0/24 [120/3] via 10.1.1.2, eth1_0, 00:00:16
C>* 192.168.1.0/24 is directly connected, eth0_0
R>* 192.168.2.0/24 [120/2] via 10.1.1.2, eth1_0, 00:00:16
R>* 192.168.3.0/24 [120/6] via 10.1.1.2, eth1_0, 00:00:16
S>* 192.168.4.0/24 [1/0] via 192.168.1.25, eth0_0
```

## RIP security

Like in other dynamic systems, the advantage of dynamic routing is that the routes are learnt automatically by routers, so the configuration tasks are limited for the network administrator, but the counterpart is that there are risks. Security problems could lead to a DoS. For instance a hacked router could announce falsified routing data that could be automatically propagated in the whole network. As RIP is an IGP, i.e. an internal protocol, other security measures could prevent this risk. However, to limit these security problems, security features have been implemented.

In this context, the advantage of RIP v2 compared to RIP v1 is that the former allows to authenticate routing information when they are transmitted between routers. Only authenticated data are allowed to be used by routers.

## RIP authentication

RIP security is based on authentication with a shared secret that can be transmitted to a broadcast area. RIP v2 supports the two authentication methods: plain-text authentication and MD5 authentication. The authentication is interface specific (scope). It means that different authentications can be defined according to the RIP interfaces. For both authentication methods (plain text or MD5), an interface specific shared secret has to be defined. The authentication keys are shared and must be the same between neighbors.

---

**Note:** This feature is supported in RIP v2 only. Plain text authentication is the default setting in every RIP v2 packet. Encrypted authentication is based on the MD5 algorithm. In this mode of authentication, the routing update carries a 128-bit message that includes the password encrypted by the MD5 algorithm. The transmitted routing information remains in clear text.

---

Except to limit error configurations consequences where a clear text password may be enough, MD5 authentication is obviously advised for security reasons.

## Filtering RIP routes

Filtering is a complementary feature used to provide a better security to RIP protocol. The concept is based on a list that contains the addresses and or prefixes allowed to be advertised or learnt amongst routing information.

1. Specify the access-list:

```
routing
  ipv4-access-list INTERNAL permit 192.168.0.0/16
  ipv4-access-list INTERNAL deny 192.168.0.0/16
  ..
```

1. Configure the distribute-list for each interface:

```
vrf main
  routing rip
    distribute-list eth0_0 out access-list INTERNAL
    distribute-list eth2_0 out access-list INTERNAL
```

## High availability

It is sometimes useful for High Availability purpose to have redundancy between two routers. In some cases, this redundancy **MUST** not be associated with load balancing, hence in case of router swap, the routing convergence time must be addressed. This will be done, without any modification to RIP itself, but rather, with configuration tuning.

The basic idea will be:

- To share a common IP address on the shared link between the two routers (and possibly a common L2 address).
- Elect a router on the link, that will be master and real owner of the IP address, the other being the slaves
- On the Master, run RIP normally
- On Slaves, run RIP in a passive mode on the shared link, so that routing table is already present in the router
- When a router comes to Master state:
  - If no L2 address is shared, send some gratuitous ARP to update ARP caches.
  - Change the RIP interface behaviour to active: it will then announce itself .

This can be achieved by using *VRMP*. In addition to IP address management the protocol will have to re-configure each RIP daemon on the fly, reproducing the same result as the following commands:

1. On the Slave(s), be in passive mode

```
vrf main
  routing rip
    passive-interface eth0_0
```

2. On the (newly) Master, re-enable interface:

```
vrf main
  routing rip
    passive-interface eth0_0
```

## OSPFv3

### OSPF v3 Overview

OSPF v3 is a redesign of OSPF v2 which adds support for a generic address family. Up to now, only the IPv6 address family has been defined. The OSPF v3 protocol is first described in **RFC 2740** (<https://tools.ietf.org/html/rfc2740.html>). It inherits most of the OSPF v2 mechanisms (Flooding, DR, LSU (Link State Update),...) with little changes.

In OSPF v3, router-id has the same format as OSPF v2, new and modified LSAs have been created to handle the flow of IPv6 addresses and prefixes in an OSPF v3 network. The new LSAs introduced in OSPF v3 are the Link LSA, and the Intra-Area-Prefix LSA.

To get more information about OSPF v2, please look at the following reference *OSPF v2*.

### OSPF v3 terminology

Most of the acronyms used for OSPF v3 are common with OSPF v2. More information at following link *OSPF v2 terminology*.

### OSPF v3 packets

OSPF v3 operates over IP protocol number 89, like with IPv4. Also, hello messages are carried over `ff02:5`. Similarly, `ff02:6` is used for messages to DR and BDR

All basic OSPF packet types can be found on OSPF v3 too. It is worth to be noted that LSA of OSPF v2 can be found on OSPF v3.

There are however some specificities:

- The link state type values are different. Router LSA type id is `0x2001` (formerly 1 in OSPF v2). Network LSA value is `0x2002`, inter-Area Prefix LSA is `0x2003` (formerly network summary LSA type 3), inter-Area Router LSA is `0x2004` (formerly ASBR summary LSA type 4), AS-external LSA type id is `0x4005` (formerly type 5), Group Membership LSA type id is `0x2006` (formerly type 6), Type-7 LSA type id is `0x2007` (formerly NSSA external LSA).
- A new link state type is available : Link LSA type id is `0x0008`. This message is dedicated to local link information only.
- Another link state type is available : The Intra-area Prefix LSA with type id value set to `0x2009`. That message is used to carry intra-area network information previously included in Network LSA used with SPF calculation. This separation permits adding or removing IP subnets without modifying the SPF tree.

## RFC

**RFC 5340** (<https://tools.ietf.org/html/rfc5340.html>): OSPF version 3

### See also:

The *OSPF v3 command reference*

## Configuring OSPF v3

- *Basic elements for configuration*
- *Verifying operation*
- *Configuration example*

### Basic elements for configuration

The configuration of OSPF v3 in a single area is similar to the configuration of OSPF v2, with slight changes. The creation of the routing instance is similar with what has been done for OSPF v2.

Here is a sample OSPF v3 configuration. OSPF v3 is activated on interfaces `eth0_0` and `eth1_0`. The interface `eth1_0` is in passive mode, which means it only emits OSPF packets and does not receive them.

```
vrf main
  routing ospf6
    router-id 10.125.0.1
    interface eth1_0 area 0.0.0.0
    interface eth0_0 area 0.0.0.0
    ..
    ..
  routing interface eth1_0
    ipv6 ospf6 passive true
```

You can disable OSPF v3 without having to remove the configuration, by using following command:

```
vrf main
  routing ospf6
    enabled false
```

Nonetheless, it is always possible to suppress OSPF v3 configuration:

```
vrf main
  del routing ospf6
  ..
```

## Verifying operation

The following commands can be used to verify OSPF v3 operation.

- Display the global OSPF parameters (timers, area, router-id, etc.):

```
vrouter> show ospf6
OSPFv3 Routing Process (0) with Router-ID 10.125.0.1
Running 00:00:44
LSA minimum arrival 1000 msec
Initial SPF scheduling delay 0 millisecond(s)
Minimum hold time between consecutive SPFs 50 millisecond(s)
Maximum hold time between consecutive SPFs 5000 millisecond(s)
Hold time multiplier is currently 1
SPF algorithm last executed 00:00:22 ago, reason L+
Last SPF duration 0 sec 40 usec
SPF timer is inactive
Number of AS scoped LSAs is 0
Number of areas in this router is 1

Area 0.0.0.0
  Number of Area scoped LSAs is 2
  Interface attached to this area: eth0_0 eth1_0
SPF last executed 22.662241s ago
```

- Display the OSPF v3 route:

```
vrouter> show ospf6 route
*N IA 2001:500:1::/64          ::          eth0_0 00:02:50
*N IA 3ffe:1::/64           ::          eth1_0 00:02:50
```

- Display the OSPF configuration for the specified interface:

```
vrouter> show ospf6 interface eth0_0
eth0_0 is up, type BROADCAST
Interface ID: 3
Internet Address:
  inet6: 2001:500:1::1/64
  inet6: fe80::dced:1ff:fe4c:9269/64
Instance ID 0, Interface MTU 1500 (autodetect: 1500)
MTU mismatch detection: enabled
Area ID 0.0.0.0, Cost 100
State DR, Transmit Delay 1 sec, Priority 1
Timer intervals configured:
  Hello 10, Dead 40, Retransmit 5
DR: 10.1.1.1 BDR: 0.0.0.0
```

(continues on next page)

(continued from previous page)

```

Number of I/F scoped LSAs is 1
 0 Pending LSAs for LSUupdate in Time 00:00:00 [thread off]
 0 Pending LSAs for LSack in Time 00:00:00 [thread off]

```

- Display the state of the relations with the neighbors:

```
vrouter> show ospf6 neighbor
```

| Neighbor ID | Pri | DeadTime | State/IfState | Duration | I/F[State] |
|-------------|-----|----------|---------------|----------|------------|
| 10.125.0.2  | 1   | 00:00:31 | Full/BDR      | 00:00:16 | eth1_0[DR] |

## Configuration example

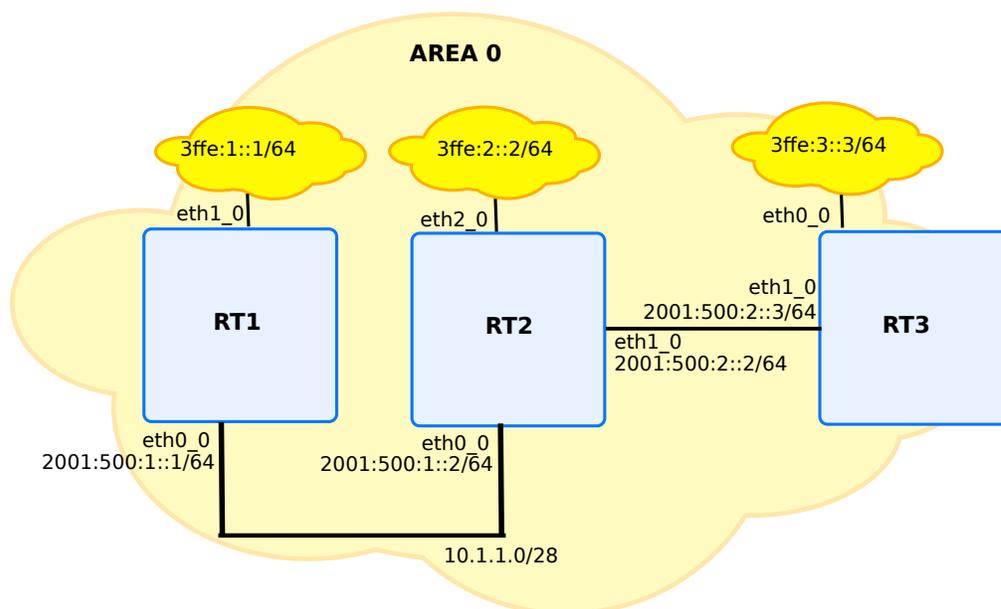


Fig. 22: Basic OSPF v3 configuration

### rt1

```

vrf main
  routing ospf6
    router-id 10.1.1.1
    interface eth0_0 area 0.0.0.0
    interface eth1_0 area 0.0.0.0
    ..

```

(continues on next page)

(continued from previous page)

```
..
interface
  physical eth1_0
    ipv6 address 3ffe:1::1/64
  ..
..
physical eth0_0
  ipv6 address 2001:500:1::1/64
  ..
..
```

**rt2**

```
vrf main
  routing ospf6
    router-id 10.1.1.2
    interface eth0_0 area 0.0.0.0
    interface eth1_0 area 0.0.0.0
    interface eth2_0 area 0.0.0.0
    ..
  ..
  interface
    physical eth2_0
      ipv6 address 3ffe:2::2/64
    ..
  ..
  physical eth0_0
    ipv6 address 2001:500:1::2/64
    ..
  ..
  physical eth1_0
    ipv6 address 2001:500:2::2/64
    ..
  ..
```

**rt3**

```
vrf main
  routing ospf6
    router-id 10.1.1.3
    interface eth0_0 area 0.0.0.0
    interface eth1_0 area 0.0.0.0
    ..
    ..
  interface
    physical eth0_0
      ipv6 address 3ffe:3::3/64
      ..
    ..
    physical eth1_0
      ipv6 address 2001:500:2::3/64
      ..
    ..
```

- Check OSPF v3 operations:

```
rt1> show ospf6 neighbor
```

| Neighbor ID | Pri | DeadTime | State/IfState | Duration | I/F[State] |
|-------------|-----|----------|---------------|----------|------------|
| 10.1.1.2    | 1   | 00:00:32 | Full/BDR      | 00:03:25 | ntfp1[DR]  |

**Note:** The state must be at Full, otherwise, this means that the OSPF v3 neighborhood is not correctly formed.

```
rt1> show ospf6 route
```

| Destination           | Gateway                  | I/F             |
|-----------------------|--------------------------|-----------------|
| *N Ia 2001:500:2::/64 | ::                       | eth1_0 00:17:01 |
| *N Ia 2001:500:1::/64 | fe80::dced:1ff:fee4:395c | eth1_0 00:16:56 |
| *N Ia 3ffe:1::/64     | ::                       | eth1_0 00:17:01 |
| *N Ia 3ffe:2::/64     | fe80::dced:1ff:fee4:395c | eth1_0 00:16:56 |
| *N Ia 3ffe:3::/64     | fe80::dced:1ff:fee4:395c | eth1_0 00:16:56 |

- Display the OSPF v3 Link-State databases and information about LSAs (Link State Advertisements)

```
vrouter> show ospf6 database
```

```
Area Scoped Link State Database (Area 0.0.0.0)
```

(continues on next page)

(continued from previous page)

| Type   | LSId    | AdvRouter | Age  | SeqNum   | Payload                  |
|--|---------|-----------|------|----------|--------------------------|
| Rtr  | 0.0.0.0 | 10.1.1.1  | 429  | 80000002 | 10.1.1.1/0.0.0.3         |
| Rtr  | 0.0.0.0 | 10.1.1.2  | 237  | 80000003 | 10.1.1.1/0.0.0.3         |
| Rtr  | 0.0.0.0 | 10.1.1.2  | 237  | 80000003 | 10.1.1.2/0.0.0.8         |
| Rtr  | 0.0.0.0 | 10.1.1.3  | 238  | 80000002 | 10.1.1.2/0.0.0.8         |
| Net  | 0.0.0.3 | 10.1.1.1  | 429  | 80000001 | 10.1.1.1                 |
| Net  | 0.0.0.3 | 10.1.1.1  | 429  | 80000001 | 10.1.1.2                 |
| Net  | 0.0.0.8 | 10.1.1.2  | 237  | 80000001 | 10.1.1.2                 |
| Net  | 0.0.0.8 | 10.1.1.2  | 237  | 80000001 | 10.1.1.3                 |
| INP  | 0.0.0.0 | 10.1.1.1  | 429  | 80000003 | 3ffe:1::/64              |
| INP  | 0.0.0.3 | 10.1.1.1  | 429  | 80000001 | 2001:500:1::/64          |
| INP  | 0.0.0.0 | 10.1.1.2  | 237  | 80000004 | 3ffe:2::/64              |
| INP  | 0.0.0.8 | 10.1.1.2  | 237  | 80000001 | 2001:500:2::/64          |
| INP  | 0.0.0.0 | 10.1.1.3  | 238  | 80000003 | 3ffe:3::/64              |
| I/F Scoped Link State Database (I/F loop in Area 0.0.0.0)  |         |           |      |          |                          |
| Type   | LSId    | AdvRouter | Age  | SeqNum   | Payload                  |
| Lnk  | 0.0.0.5 | 10.1.1.1  | 1116 | 80000001 | fe80::4426:67ff:fe5:88b4 |
| I/F Scoped Link State Database (I/F ntfp1 in Area 0.0.0.0) |         |           |      |          |                          |
| Type   | LSId    | AdvRouter | Age  | SeqNum   | Payload                  |
| Lnk  | 0.0.0.3 | 10.1.1.1  | 1109 | 80000001 | fe80::dced:1ff:fe4c:9269 |
| Lnk  | 0.0.0.6 | 10.1.1.2  | 432  | 80000001 | fe80::dced:1ff:fee4:395c |
| AS Scoped Link State Database                              |         |           |      |          |                          |
| Type   | LSId    | AdvRouter | Age  | SeqNum   | Payload                  |

### Configuring OSPF v3 in multiple areas

Like in IPv4 with OSPF v2, OSPF v3 permits the use of multiple areas, and an OSPF v3 router may be an ABR; that is to say that it is a router having at least one interface in one area and another interface in a different area.

### OSPF v3 stub area overview

OSPF v3 implement supports stub area like with OSPF v2. Below example illustrates how to declare area 1 as a stub area.

```
vrf main
  routing ospf6
    area 1 stub
```

### Totally stubby area overview

OSPF v3 implement supports totally stub area like with OSPF v2. Below example illustrates how to declare area 1 as a totally stubby area.

```
vrf main
  routing ospf6
    area 1 stub summary false
```

### OSPF v3 options

Below is given some illustration that help on how to configure OSPF v3.

#### OSPF v3 cost

Following example sets the interface output cost. If not set, the value is automatically calculated based on the bandwidth of the interface. By default, cost is set to 1 for a 100MB link.

```
vrf main
  routing interface eth0_0
    ipv6 ospf6 cost 20
```

As said before, if not explicitly set, the cost is determined by the bandwidth of the interface. It is possible to impact the cost value by changing the default reference bandwidth used. By default, it is 100MB. Below example illustrates a reference of 1GB.

```
vrf main
  routing ospf6
    auto-cost 1000
```

### OSPF v3 priority

The interface's router Priority for election of designated router can be modified, by using following command on routing interface mode.

```
vrf main
  routing interface eth0_0
    ipv6 ospf6 priority 10
```

Default value is 1.

### OSPF v3 hello interval

Below example illustrates how to set interval for hello messages, per interface.

```
vrf main
  routing interface eth0_0
    ipv6 ospf6 hello-interval 20
```

Default value is 10 seconds.

### OSPF v3 transmit-delay

Below example illustrates how to configure per interface transmit-delay:

```
vrf main
  routing interface eth0_0
    ipv6 ospf6 transmit-delay 3
```

Default value is 1.

### Passive interface

This feature should be used when it is required to prevent some router's interfaces from forming OSPF adjacencies. It may be for instance to include a subnet into the OSPF routing process (and LSD), without actually running OSPF on the interface of the router connected to that subnet. This is useful to announce stub networks instead of external LSA. This is particularly adapted for interfaces that are used as BGP peering links or for customer connectivity.

```
vrf main
  routing interface eth0_0
    ipv6 ospf6 passive true
  ..
  ..
```

(continues on next page)

(continued from previous page)

```
routing ospf6
  interface eth0_0 area 0.0.0.0
  ..
  ..
```

## ECMP

There might be some situation in which, for a common destination, OSPF has different paths, of equal cost to reach that destination. In such situation, network traffic may be distributed equally among all the equal cost paths. This situation relies on the ECMP capabilities.

## BFD In OSPF v3

With BFD usage in OSPF v3, the failover mechanism is greatly improved by detecting the loss of remote OSPF v3 neighbors. Instead of relying on standard hello mechanisms, BFD permits faster convergence. To get more information on BFD, please see *BFD*.

## BFD Configuration And Monitoring In OSPF v3

A BFD peer session context is created, along with discovering OSPF v3 neighbors. Due to the nature of OSPF v3, all created BFD peer contexts are single-hop, and are based on IPv6.

```
vrf customer1
  routing ospf6
    router-id 10.125.0.1
    interface eth1_0 area 0.0.0.1
    .. ..
  routing interface eth1_0
    ipv6 ospf6 track bfd
```

Then you can continue the configuration as usual. For timer settings, the default emission and reception settings are set to 300000 microseconds, which may not be what is wished. In that case, it is possible to override default timers, by configuring general timer settings. More information is given in *Configuring general BFD settings*.

```
vrouters> show ospf6 vrf customer1 interface eth1_0
eth1_0 is up, type BROADCAST
Interface ID: 4
Internet Address:
  inet6: 2001:db8:4::2/64
  inet6: fe80::20e2:2bff:fe5c:d44b/64
Instance ID 0, Interface MTU 1500 (autodetect: 1500)
```

(continues on next page)

(continued from previous page)

```

MTU mismatch detection: enabled
Area ID 0.0.0.1, Cost 10
State BDR, Transmit Delay 1 sec, Priority 1
Timer intervals configured:
  Hello 10, Dead 40, Retransmit 5
DR: 10.254.254.4 BDR: 10.254.254.2
Number of I/F scoped LSAs is 2
  0 Pending LSAs for LSUpdate in Time 00:00:00 [thread off]
  0 Pending LSAs for LSack in Time 00:00:00 [thread off]
BFD: Detect Multiplier: 3, Min Rx interval: 300, Min Tx interval: 300

```

```

vrouters> show ospf6 vrf customer1 neighbor
Neighbor ID      Pri    DeadTime    State/IfState      Duration I/F[State]
10.125.0.2       1      00:00:30    Full/DR            00:22:34 eth1_0[BDR]

```

```

vrouters> show bfd vrf customer1 session single-hop destination_
↪ fe80::347c:8fff:fe10:e2b4
BFD Peer:
  peer fe80::347c:8fff:fe10:e2b4 local-address fe80::bcda:24ff:fef7:38d3 interface_
↪ eth1_0
  ID: 322201613
  Remote ID: 2746639856
  Status: up
  Uptime: 9 minute(s), 49 second(s)
  Diagnostics: ok
  Remote diagnostics: ok
  Local timers:
    Receive interval: 600ms
    Transmission interval: 600ms
    Echo transmission interval: 50ms
  Remote timers:
    Receive interval: 300ms
    Transmission interval: 300ms
    Echo transmission interval: 50ms

```

## RIPNG

### Overview

RIPNG is the equivalent of RIP, but for IPv6 networks. It uses the Bellman-Ford algorithm, and as RIP, the network diameter is limited to 15 hops. It is described by the IETF **RFC 2080** (<https://tools.ietf.org/html/rfc2080.html>), it is a RIP v2 redesign that supports the 128 bit IPv6 addresses. It uses multicast UDP on the well-known group ff02::9 and port 521. Due to the IPSEC requirement of IPv6 stacks, RIPNG does not have the security features that RIP v2 provides: it has to be handled by the IPv6 security layer (IPSEC).

### RFC

**RFC 2080** (<https://tools.ietf.org/html/rfc2080.html>) RIPng for IPv6

### RIPng Configuration

#### Basic elements for configuration

Starting RIPNG can be done by using a very simple configuration. Example below illustrates a basic configuration setup with one network configured. Automatically, RIPNG will operate over all the interfaces where an IP address is defined, whose network address is included in the provided network prefix. Network addresses included in this prefix and defined on these interfaces will be advertised.

It is worth to be noted that RIPNG does not announce the link-local prefixes (fe80::).

```
vrf main
  routing ripng
    network 2001::/16
  ..
  ..
  commit
```

As mentioned in above config, RIPNG is activated, with providing network prefix. It is also possible to provide interface name. If an interface name is provided, RIPNG will then be activated on this interface and all IPv6 network prefixes defined on this interface will be advertised.

```
vrf main
  routing ripng
    interface eth1_0
```

RIPNG can be stopped by using following command:

```
vrf main
  del routing ripng
  commit
```

Alternatively, it is also possible to just disable RIPNG without having to remove the whole configuration.

```
vrf main
  routing ripng enabled false
  commit
```

Currently, RIPNG is only supported in VRF main.

## Verifying RIPng configuration

The following commands can be used to verify RIPNG operation.

### show ripng

This command displays the RIB of the RIPNG protocol.

```
vrouter> show ripng
Codes: R - RIPng, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
  (n) - normal, (s) - static, (d) - default, (r) - redistribute,
  (i) - interface, (a/S) - aggregated/Suppressed

  Network          Next Hop          Via      Metric Tag Time
R(n) fec0:1::/64   fe80::dced:3ff:fe4a:8933 ntfp3    2    0 02:39
C(i) fec0:2::/64   ::                self     1    0
```

### show ripng status

This command displays Turbo Router running state of RIPNG.

```
vrouter> show ripng status
Routing Protocol is "RIPng"
  Sending updates every 30 seconds with +/-50%, next due in 2 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Redistributing:
```

(continues on next page)

(continued from previous page)

```

Default version control: send version 1, receive version 1
  Interface      Send  Recv
  eth1_0         1    1
Routing for Networks:
  eth1_0
Routing Information Sources:
  Gateway                BadPackets  BadRoutes  Distance  Last Update
  fe80::dced:3ff:fe4a:8933    0           0          120       00:00:04

```

## RIPng configuration options

Like RIP, RIPNG has many options, besides with RIPNG there is a possibility to aggregate the addresses and to declare one network, these options are described in detail in the following sections.

### Split horizon

Like RIP, RIPNG does not announce the learnt prefixes on the interfaces from which they were learnt. This is the default behavior of Turbo Router.

To disable the split horizon feature on a given interface type the following command at the interface level of the routing context.

```

vrf main
  routing interface eth0_0
    ipv6 ripng split-horizon disabled
  ..
  ..
  ..

```

To come back to default behavior and enable split-horizon, use the following command:

```

vrf main
  routing interface eth0_0
    ipv6 ripng split-horizon simple
  ..
  ..
  ..

```

## Split horizon with poisoned reverse

The goal of poisoning the reverse path is to increase the convergence of the RIPNG algorithm to quickly kill the RIPNG routing loops. When split-horizon with poisoned reverse path is enabled, the prefixes which are learnt via an interface are announced back each 30 seconds with a metric of 16 (i.e. infinite).

This option is configured at the interface level at the routing context by typing the following command at the interface level of the routing context.

```
vrf main
  routing interface eth0_0
    ipv6 ripng split-horizon poisoned-reverse
    ..
  ..
..
```

- Disable poisoned-reverse:

```
vrf main
  routing interface eth0_0
    ipv6 ripng split-horizon simple
    ..
  ..
..
```

This will disable the poisoned-reverse option in the RIPNG configuration and remain in the split-horizon RIPNG policy.

## Default route advertisement

The default-information originate command can be used to allow RIPNG to advertise the default route (::/0).

```
vrf main
  routing ripng
    default-information-originate true
    ..
  ..
..
```

When enabling this option, default route will be displayed in the list of entries that RIPNG displays:

```
vrouter> show ripng
Codes: R - RIPng, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
```

(continues on next page)

(continued from previous page)

(i) - interface, (a/S) - aggregated/Suppressed

| Network   | Next Hop | Via  | Metric | Tag | Time |
|-----------|----------|------|--------|-----|------|
| R(d) ::/0 | ::       | self | 1      | 0   |      |

**Note:** When a router is advertising a default route, it is advised that it is itself configured with its own default IPv6 route to avoid it becomes a blackhole:

```
vrf main
  routing static
    ipv6-route 0::0/0 next-hop eth1_0
```

## Static RIPng route

The RIPNG process can announce a route that has no origin. It means that it has not been introduced into the RIPNG RIB by the redistribute command.

- Add a route to the RIPNG RIB (in the RIPNG context):

```
vrf main
  routing ripng
    static-route 2003::/64
  ..
  ..
```

This static route appears in the RIB with the R(s) tag.

```
vrouter> show ripng
Codes: R - RIPng, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
  (n) - normal, (s) - static, (d) - default, (r) - redistribute,
  (i) - interface, (a/S) - aggregated/Suppressed

Network      Next Hop      Via      Metric Tag Time
R(s) 2001::/64  ::          self      1      0
```

It is announced as RIPNG route but with the subcode (s) which means that the prefix was learned by a static route. With this command, a black-hole could be announced.

## Manage the redistributed metrics

Since the routing protocols are not the same (BGP, static, connected), the associated metrics cannot be compared. An administrative distance, that is composed of hop count, can be set with the following command into the RIPNG context.

```
vrf main
  routing ripng
    redistribute connected metric 3
    redistribute static metric 4
    redistribute bgp metric 8
  ..
..
```

---

**Note:** Due to the maximum RIPNG metric of 16, these commands decrease the size of your network.

---

The default redistribution metric into RIPNG is 1.

## Modify timers

The routing protocols are based on many timers that control the stability of your network and the time convergence of the algorithms. RIPNG is based on three timers:

- a. The routing table update in seconds: default 30 s,
  - b. The routing information timeout in seconds: default 180 s.,
  - c. The garbage collection in seconds: default 120 s.
- Change the default timers:

```
vrf main
  routing ripng
    timers update-interval 30 holddown-interval 180 flush-interval 120
  ..
..
```

- Check the timers values:

```
vrouter> show ripng status
Routing Protocol is "RIPng"
  Sending updates every 30 seconds with +/-50%, next due in 4 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  [...]
```

---

**Note:** Do not change any default value if you are deploying a RIPNG network over a LAN. They should be changed only over some very low bandwidth links (about 32 Kbit/s or less) or over the cost expensive links.

---

## Route aggregation

The routes redistributed by RIPNG can be aggregated to decrease the FIB table or to hide the internal architecture of your network. This aggregation can be done with the following command:

- Aggregate routes:

```
vrf main
  routing ripng
    aggregate 3ffe:501:ffff:4000::/52
  ..
..
```

---

**Note:** This feature is specific to RIPNG and is not available with RIP.

---

## Example

```
vrf main
  interface
    physical eth0_0
      ipv6 address 3ffe:501:ffff:4001::4/64
    ..
    physical eth1_0
      ipv6 address 3ffe:501:ffff:4000::4/64
    ..
    physical eth2_0
      ipv6 address 3ffe:501:ffff:1::4/64
    ..
  ..
  routing ripng
    aggregate-address 3ffe:501:ffff:4000::/52
    network 3ffe:501:ffff::/48
  ..
..
```

It leads to the following RIPNG RIB:

```

vrouters> show ripng
Codes: R - RIPng, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
  (n) - normal, (s) - static, (d) - default, (r) - redistribute,
  (i) - interface, (a/S) - aggregated/Suppressed

  Network      Next Hop      Via      Metric Tag Time
C(i) 3ffe:501:ffff:1::/64
      ::        self        1      0
R(a) 3ffe:501:ffff:4000::/52
      ::        self        1      0
C(Si) 3ffe:501:ffff:4000::/64
      ::        self        1      0
C(Si) 3ffe:501:ffff:4001::/64
      ::        self        1      0
    
```

The tag R(a) means that the prefix 3ffe:501:fff:4000::/52 is an aggregated one.

### RIPNG options example

In this example we will configure 4 routers rt1, rt2, rt3 and rt4 to support the RIPNG options.

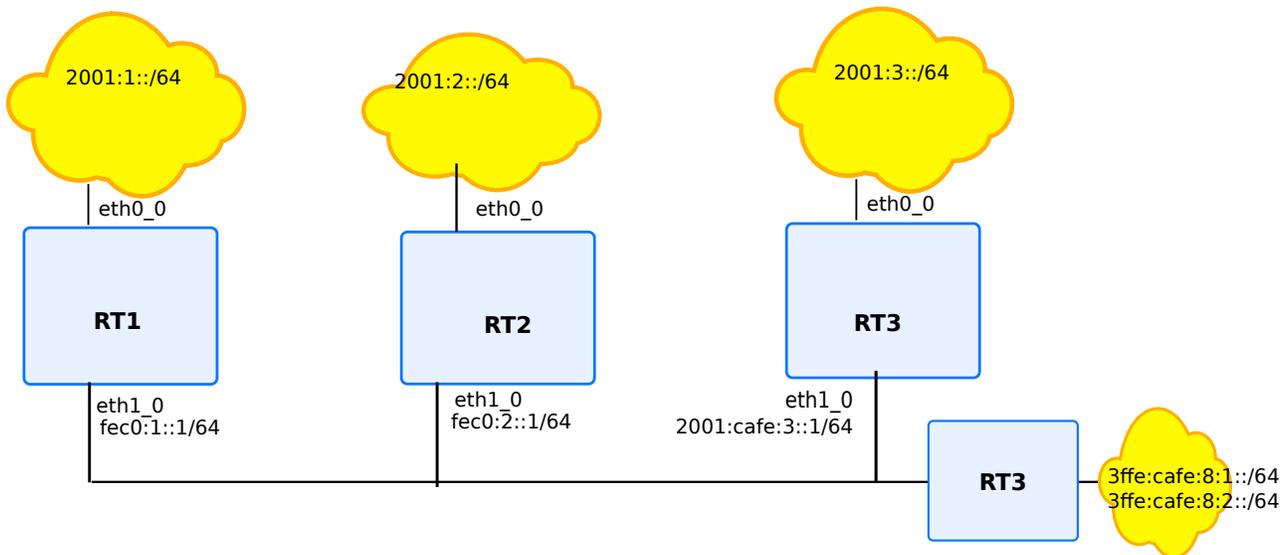


Fig. 23: RIPNG options configuration example

The required features are being tested:

- rt1: RIPNG static route
- rt2: Split-horizon, poison-reverse
- rt3: redistribute connected
- rt4: modify timers option, aggregate address option, default route information

**rt1**

```
vrf main
interface
  physical eth0_0
    ipv6 address 2001:1::1/64
    ..
  physical eth1_0
    ipv6 address fec0:1::1/64
    ..
  ..
routing ripng
  network 2001::/16
  network fec0:1::/64
  static-route fec0:1::/16
  ..
..
```

**rt2**

```
vrf main
interface
  physical eth0_0
    ipv6 address 2001:2::1/64
    ..
  physical eth1_0
    ipv6 address fec0:2::1/64
    ..
  ..
routing
  ripng
    network 2001::/16
    ..
  ..
interface eth0_0
  ipv6 ripng split-horizon disable
```

(continues on next page)

(continued from previous page)

```
..
interface eth1_0
  ipv6 ripng split-horizon poisoned-reverse
..
..
```

**rt3**

```
vrf main
interface
  physical eth0_0
    ipv6 address 2001:3::1/64
  ..
  physical eth1_0
    ipv6 address 2001:cafe:3::1/64
  ..
..
routing ripng
  network 2001::/16
  redistribute connected metric 5
..
..
```

**rt4**

```
vrf main
routing ripng
  aggregate 3ffe:cafe:8::/48
  default-information-originate true
  network 3ffe:cafe:8:1::/64
  network 3ffe:cafe:8:2::/64
  timers update-interval 30 holddown-interval 180 flush-interval 90
..
..
```

## BFD

### BFD Overview

Bidirectional Forwarding Detection is a network protocol that permits low overhead and rapid detection of changes in paths reachability between two network devices.

There was a need to have a replacer for other keepalive and hello mechanisms provided by other routing protocols. Actually, BFD detects faster failures, than those mentioned mechanisms, and as such it becomes a mandatory requirement in today deployments.

BFD principle consists in exchanging specific packets with remote peer. As such, it is needed to configure both endpoints with BFD. The rate of emission and failover criterium are embedded in the packets. Based on the non reception of packets, the BFD endpoint will accordingly detect a failover with remote endpoint.

The protocol has improved along the years, and became a standard, from 2011. Initially, protocol was supporting only connected links, with `single-hop`. Now, BFD is able to monitor non directly connected links, with the `multi-hop`. BFD can also work in `echo-mode`. Both IPv4 or IPv6 links can be monitored.

BFD notifies the user about the reachability of such paths, and can also interact with other routing protocols. This is the case with BGP, where neighbors can be monitored by using BFD. This is also the case with OSPF and OSPF v3. As such, BFD notifies daemons of the rapid change on path reachability, and as consequence, routing protocols update routing tables quicker.

### BFD Packets

BFD operates over UDP protocol. Destination port 3784 is used by BFD `single-hop`, while 4784 port is used by BFD `multi-hop`. `echo-mode` uses 3785 port. Moreover, the source port range is limited by the standard, as it can operate over the range from 49152 to 65535.

The BFD control packets payload contains some fields that determine how the BFD operates. For instance, if `echo-mode` is used, a field indicates that echo mode is used. It contains a discriminator ID, that is locally generated and determines the BFD session itself. the remote discriminator of remote endpoint is also mentioned in the BFD packet.

As mentioned before, BFD operates on time constraints. Those time constraints are chosen, after exchanging between both endpoints. The timer constraints are encoded in the BFD control packet. For instance, the local endpoint indicates the desired received interval that the remote endpoint can use to send BFD control packets. Reversely, the desired transmitted interval is also encoded in the packet.

## BFD Operation

The main operation of BFD is to detect the quickest possible the loss of a remote peer. The detection time is calculated independently in each direction by the receiving system based on the negotiated transmit interval and the detection multiplier. For instance, if the agreed transmit interval is set to 100 ms, and the detection multiplier is set to three, the timeout calculation will be around 300 ms.

## RFC

The BFD is handled by FRR (<https://frrouting.org/>). Following features are provided:

**RFC 5880** (<https://tools.ietf.org/html/rfc5880.html>): Bidirectional Forwarding Detection (BFD)

**RFC 5881** (<https://tools.ietf.org/html/rfc5881.html>): Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)

**RFC 5882** (<https://tools.ietf.org/html/rfc5882.html>): Generic Application of Bidirectional Forwarding Detection (BFD)

**RFC 5883** (<https://tools.ietf.org/html/rfc5883.html>): Bidirectional Forwarding Detection (BFD) for Multihop Pathq

### See also:

The *command reference* for details.

## BFD Configuration

There is a list of necessary elements to know when forging a BGP configuration.

- *Basic Elements For Configuring BFD Entry*
- *Basic Elements For General Configuration*
- *Basic Elements For Monitoring*
- *Configuration With Remote Daemons*

### Basic Elements For Configuring BFD Entry

When forging a BFD configuration, the destination IP and the kind of BFD variant determine a BFD session.

```
tracker bfd main type single-hop address 10.125.0.2 vrf main
```

Three additional parameters determine the BFD session: the source address, the interface name and the vrf name. The source and interface options permit to stick with routing constraints.

```
tracker bfd other type single-hop address 10.125.0.2 source 10.125.0.1 vrf main
```

BFD provides low overhead. However, it provides a per peer custom configuration, that permits lowering (or increasing) the timers that determine how, and when BFD packets are sent, and received.

```
tracker bfd othername
  type single-hop
  address 10.125.0.2
  vrf main
  detection-multiplier 6
  required-receive-interval 600000
  desired-transmission-interval 600000
```

It is possible to disable bfd session usage, by using following command. Note that you will have to check that no other daemon is using BFD. Otherwise, the command will not be successful.

```
del tracker bfd othername
```

## Basic Elements For General Configuration

It is possible to change general timer settings that will apply to the BFD sessions automatically created by routing protocols (like BGP). This facility avoids the heavy task to configure for each session the newly wished parameters. Note that configured values are expressed in microseconds.

```
routing bfd
  detection-multiplier 7
  required-receive-interval 800000
  desired-transmission-interval 200000
```

Reversely, it is possible to revert to default settings. By default, detect multiplier is set to 3, while default required-receive-interval and transmit-interval is set to 300 milliseconds.

```
routing bfd
  del detection-multiplier
  del required-receive-interval
  del desired-transmission-interval
```

## Basic Elements For Monitoring

You can use the `show bfd` commands to watch for BFD sessions.

Following commands gives detailed BFD information about the BFD sessions status and statistics.

```
vrouter> show bfd vrf main session single-hop destination 10.125.0.2
peer 10.125.0.2 singlehop local-address 10.125.0.1
  ID: 2916604864
  Remote ID: 1159562547
  Status: up
  Uptime: 37 second(s)
  Diagnostics: ok
  Remote diagnostics: ok
  Local timers:
  Receive interval: 300ms
  Transmission interval: 300ms
  Echo transmission interval: 50ms
  Remote timers:
  Receive interval: 300ms
  Transmission interval: 300ms
  Echo transmission interval: 50ms
```

```
vrouter> show bfd vrf main sessions counters
BFD Peers:
peer 10.125.0.2 singlehop local-address 10.125.0.1
Control packet input: 182 packets
Control packet output: 181 packets
Echo packet input: 0 packets
Echo packet output: 0 packets
Session up events: 1
Session down events: 0
Zebra notifications: 2
```

## Configuration With Remote Daemons

In addition to be able to create BFD peer sessions by using nc-cli of bfd, it is possible to dynamically create BFD peer sessions by relying on remote daemons.

### See also:

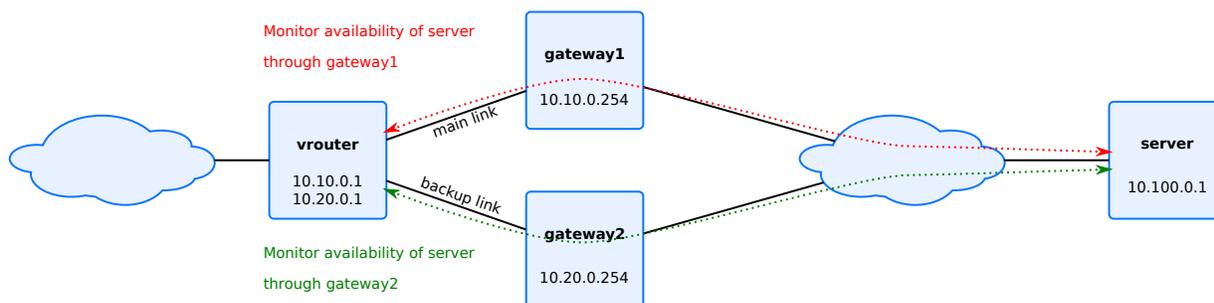
- Using BFD with BGP, see *Configuring BGP with BFD*.
- Using BFD with OSPF, see *Configuring OSPF with BFD*.
- Using OSPF v3 with BGP, see *Configuring OSPFv3 with BFD*.

- Using BFD with static routes, see *Configuring Static Routes with BFD*.

## Path Monitoring

The tracker service provides helpers to monitor the availability of IP addresses, using ICMP echo requests.

In the following example, the router has two links to reach the server: the main link and the backup link. Trackers can be used to monitor the availability of the server through both links, and configure static routing accordingly. A higher priority is assigned to the main link, using the distance parameter in the static routing context.



This can be configured as below:

```
vrouter running config# / tracker
vrouter running tracker# icmp main vrf main address 10.100.0.1 gateway 10.10.0.254
↪source 10.10.0.1
vrouter running tracker# icmp backup vrf main address 10.100.0.1 gateway 10.20.0.254
↪source 10.20.0.1
vrouter running tracker# / vrf main routing static
vrouter running static# ipv4-route 10.100.0.0/16
vrouter running ipv4-route 10.100.0.0/16#! next-hop 10.10.0.254 track main distance 1
vrouter running ipv4-route 10.100.0.0/16# next-hop 10.20.0.254 track backup distance 2
```

To display the trackers state:

```
vrouter running config# / tracker
vrouter running tracker# show state
tracker
  icmp main address 10.100.0.1 vrf main source 10.10.0.1 gateway 10.10.0.254 period
↪500 threshold 1 total 1 discriminator 583249321 state down diagnostic timeout type
↪icmp-echo
  icmp backup address 10.100.0.1 vrf main source 10.20.0.1 gateway 10.20.0.254
↪period 500 threshold 1 total 1 discriminator 489368122 state up diagnostic ok type
↪icmp-echo
```

(continues on next page)

(continued from previous page)

..

The same configuration can be made using this NETCONF XML configuration:

```

ubuntu1804 running config# show config xml
<config xmlns="urn:6wind:vrouter">
  <tracker xmlns="urn:6wind:vrouter/tracker">
    <icmp xmlns="urn:6wind:vrouter/tracker/icmp">
      <name>main</name>
      <vrf>main</vrf>
      <address>10.100.0.1</address>
      <gateway>10.10.0.254</gateway>
      <source>10.10.0.1</source>
      <period>500</period>
      <threshold>5</threshold>
      <total>10</total>
      <packet-size>100</packet-size>
      <packet-tos>192</packet-tos>
      <timeout>500</timeout>
    </icmp>
    <icmp xmlns="urn:6wind:vrouter/tracker/icmp">
      <name>backup</name>
      <vrf>main</vrf>
      <address>10.100.0.1</address>
      <gateway>10.20.0.254</gateway>
      <source>10.20.0.1</source>
      <period>500</period>
      <threshold>5</threshold>
      <total>10</total>
      <packet-size>100</packet-size>
      <packet-tos>192</packet-tos>
      <timeout>500</timeout>
    </icmp>
  </tracker>
  <vrf>
    <name>main</name>
    <routing xmlns="urn:6wind:vrouter/routing">
      <static>
        <ipv4-route>
          <destination>10.100.0.0/16</destination>
          <next-hop>
            <next-hop>10.10.0.254</next-hop>
            <track>main</track>
            <distance>1</distance>
          </next-hop>
        </ipv4-route>
      </static>
    </routing>
  </vrf>
</config>

```

(continues on next page)

(continued from previous page)

```

        </next-hop>
        <next-hop>
            <next-hop>10.20.0.254</next-hop>
            <track>backup</track>
            <distance>2</distance>
        </next-hop>
    </ipv4-route>
</static>
</routing>
<network-stack xmlns="urn:6wind:vrouter/system">
    <icmp/>
    <ipv4/>
    <ipv6/>
    <neighbor/>
    <contrack/>
</network-stack>
<interface xmlns="urn:6wind:vrouter/interface"/>
<logging xmlns="urn:6wind:vrouter/logging"/>
</vrf>
</config>

```

**See also:**

- The *ICMP tracker commands reference*.
- The *static routing commands reference*.

**Policy-based routing**

Policy-based routing (for IPv4 and IPv6) is a way to forward packets based on multiple criteria, not only the IP destination.

For that a set of policy routing rules is created. Each policy routing rule consists of a match (source address, input interface, protocol ...) and an action predicate (lookup in a specific table, nat ...). The rules are scanned in order of decreasing precedence. As soon as the packet matches a rule its action is performed.

Only a subset of policy-based routing options are provided. These options are:

- key:
  - priority of the rule (high number means lower priority)
- match:
  - source: source address or prefix
  - destination: destination address or prefix
  - mark: filter for the packet firewall mark

- inbound-interface: input interface
- not: flag that inverts the match result
- action:
  - lookup: longest prefix match lookup in a routing table

To add a policy-based routing rule, do:

```
vrouter running config# vrf main
vrouter running vrf main# routing policy-based-routing
vrouter running policy-based-routing# ipv4-rule 5 match source 192.15.24.0/24 action
↳lookup 12
vrouter running policy-based-routing# ipv4-rule 6 not match destination 192.168.0.0/16
↳action lookup 14
vrouter running static# commit
Configuration applied.
```

To display the policy-based routing state:

```
vrouter running config# show state vrf main routing policy-based-routing
policy-based-routing
  ipv4-rule 0 action lookup local
  ipv4-rule 5 match source 192.15.24.0/24 action lookup 12
  ipv4-rule 6 not match destination 192.168.0.0/16 action lookup 14
  ipv4-rule 32766 action lookup main
  ipv4-rule 32767 action lookup default
  ipv6-rule 0 action lookup local
  ipv6-rule 32766 action lookup main
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running config# show config xml absolute vrf main routing policy-based-routing
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <routing xmlns="urn:6wind:vrouter/routing">
      <policy-based-routing xmlns="urn:6wind:vrouter/pbr">
        <ipv4-rule>
          <priority>5</priority>
          <match>
            <source>192.15.24.0/24</source>
          </match>
          <action>
            <lookup>12</lookup>
          </action>
        </ipv4-rule>
      </policy-based-routing>
    </routing>
  </vrf>
</config>
```

(continues on next page)

(continued from previous page)

```

</ipv4-rule>
<ipv4-rule>
  <priority>6</priority>
  <not>
    <match>
      <destination>192.168.0.0/16</destination>
    </match>
  </not>
  <action>
    <lookup>14</lookup>
  </action>
</ipv4-rule>
</policy-based-routing>
<static/>
</routing>
<interface xmlns="urn:6wind:vrouter/interface"/>
</vrf>
</config>

```

**Example** The following configuration allows to forward packets to subnet 192.165.1.0/24 through different interfaces. Packets from subnet 192.168.1.0/24 are forwarded through eth0, other packets through eth1.

```

vrouter running config# vrf main
vrouter running vrf main# interface physical eth0
vrouter running physical eth0#! port pci-b0s8
vrouter running physical eth0# ipv4 address 10.125.0.2/24
vrouter running physical eth0# .. ..
vrouter running vrf main# interface physical eth1
vrouter running physical eth1#! port pci-b0s7
vrouter running physical eth1# ipv4 address 10.175.0.2/24
vrouter running physical eth1# .. ..
eth0 and eth1 physical interfaces are now configured
vrouter running vrf main# routing static
vrouter running static# ipv4-route 192.165.1.0/24 next-hop 10.175.0.2
vrouter running static# table 100 ipv4-route 192.165.1.0/24 next-hop 10.125.0.2
2 rules to forward packets to 192.165.1.0/24 are created, the first one in
the main route table via eth1, the second one in the table 100 via eth0
vrouter running vrf main# routing policy-based-routing
vrouter running policy-based-routing# ipv4-rule 5 match source 192.168.1.0/24 action.
↳lookup 100
A policy-based routing rule is added to indicate that packets from
192.168.1.0/24 must apply routes defined in table 100 (if no route is found
the routes defined in the main table will be applied)
vrouter running static# commit

```

(continues on next page)

(continued from previous page)

|                        |
|------------------------|
| Configuration applied. |
|------------------------|

**See also:**

The *command reference* for details.

### 3.1.8 QoS

#### Rate limiting

The traffic received and sent on network interfaces can be rate limited in order to prevent the device or the network to be overloaded, or to enforce maximum bit rate agreements.

Rate limiting is available on all physical and logical interfaces, in both ingress and egress of the device.

#### Rate limiting algorithm

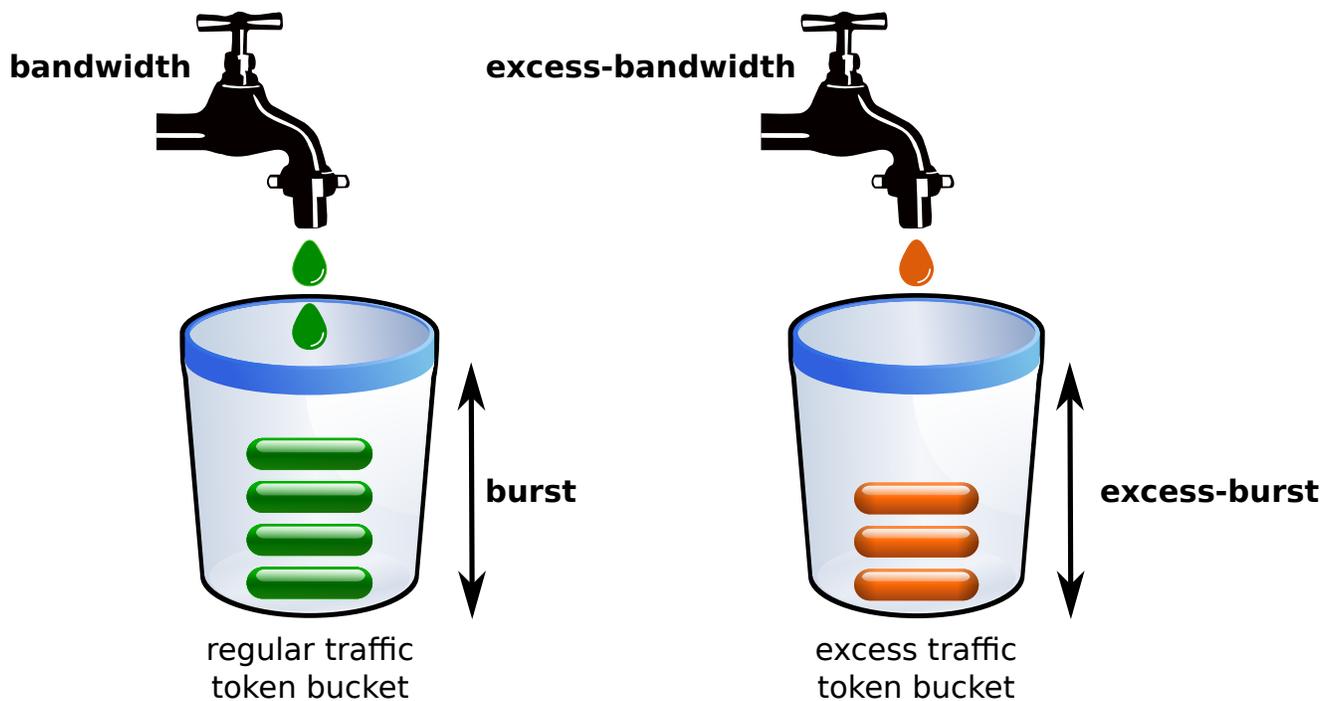
The rate limit of an interface is controlled by a policer, in charge of dropping traffic that does not fulfill a given traffic profile.

The policer specifies the maximum committed bandwidth of the regular traffic. It may optionally specify an authorized excess bandwidth, to accommodate temporary excess use.

- the traffic profile is measured by a three-color marker (see **RFC 4115** (<https://tools.ietf.org/html/rfc4115.html>)), composed of a token bucket for regular traffic and an optional token bucket for excess traffic.
- packets are then either granted access or dropped, whether they conform to the traffic profile or not:
  - if a packet fulfills the bandwidth/burst specification (green packet), it can pass.
  - else if the excess-bandwidth is non-zero and the packet fulfills the excess-bandwidth/excess-burst specification (yellow packet), it can pass.
  - otherwise the packet is out of profile (red packet), it is dropped.

Up to 4 parameters may be defined:

- **bandwidth**: maximum frame bit rate of regular traffic, a.k.a. CIR (Committed Information Rate), in bits per second (mandatory),
- **burst**: maximum burst size of regular traffic, a.k.a. CBS (Committed Burst Size), in bytes (defaults to bandwidth/80, so that the system is able to handle a burst of 100 ms at the targeted bandwidth),
- **excess-bandwidth**: maximum frame bit rate of excess traffic, a.k.a. EIR (Excess Information Rate), in bits per second (default 0),
- **excess-burst**: maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes (defaults to bandwidth/80, so that the system is able to handle a burst of 100 ms at the targeted bandwidth).



Rate limiting can be configured in two ways:

- a dedicated policer is attached to an interface ingress or egress,
- a shared policer is created, then several interfaces may bind their ingress or egress to this shared policer. All interfaces bound to this shared policer consume tokens of the same three-color marker.

## Policer templates

Policer templates are created in the global qos context with the `policer` command. They can then be referenced by interfaces or by shared policers.

Enter the global qos context:

```
vrouter running config# qos
vrouter qos#
```

Create a policer template with no authorized excess traffic:

```
vrouter running config# qos
vrouter running qos#
vrouter running qos# policer pol1
vrouter running policer pol1#! bandwidth 1G
vrouter running policer pol1# burst 2K
vrouter running policer pol1# ..
vrouter running qos#
```

Interfaces that use this policer will have their frame rate limited to 1 Gbps, with bursts up to 2 Kbytes. Frames that would cause this profile to be exceeded will be dropped.

Create a policer template with authorized excess traffic:

```
vrouter running qos# policer pol2
vrouter running policer pol2#! bandwidth 2G
vrouter running policer pol2# excess-bandwidth 15M
vrouter running policer pol2# ..
```

Interfaces that use this policer will have their frame rate limited to 2 Gbps, with bursts up to the default bandwidth/80 bytes. Excess traffic is authorized up to 15 Mbps with bursts up to the default excess-bandwidth/80 bytes. Frames that would cause this profile to be exceeded will be dropped.

Show the qos configuration:

```
vrouter running qos# show config
qos
  policer pol1
    bandwidth 1G
    burst 2K
    excess-bandwidth 0
    ..
  policer pol2
    bandwidth 2G
    excess-bandwidth 15M
    ..
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running config# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <qos xmlns="urn:6wind:vrouter/qos">
    <policer>
      <name>pol1</name>
      <burst>2000</burst>
      <excess-bandwidth>0</excess-bandwidth>
      <bandwidth>1000000000</bandwidth>
    </policer>
    <policer>
      <name>pol2</name>
      <excess-bandwidth>15000000</excess-bandwidth>
      <bandwidth>2000000000</bandwidth>
    </policer>
  </qos>
</config>
```

---

**Note:** The `policer` command defines traffic profile templates. They can be used by one or more network interfaces or shared-policers. Each use of a `policer` instantiates a new three color marker.

---

**Note:** Bandwidth and burst values can be typed as plain integers (e.g. 2000000), or with a standard power-of-1000 multiplier letter to write the value in a more compact way (e.g. 2M):

- K (for kilo): multiply by 1000
- M (for mega): multiply by 1000<sup>2</sup>
- G (for giga): multiply by 1000<sup>3</sup>
- T (for tera): multiply by 1000<sup>4</sup>

The output of `show config` and `show state` will always use the most compact form (e.g. 2M, regardless if you typed 2M, 2000K or 2000000).

This compact notation is only used in the CLI. The NETCONF XML configuration uses plain integers.

---

## Shared Policers

Shared policer are created in the global qos context with the `shared-policer` command. They can then be referenced by interfaces.

Enter the global qos context:

```
vrouter running config# qos
vrouter qos#
```

Create a policer template with no authorized excess traffic, as explained in the previous section:

```
vrouter running config# qos
vrouter running qos#
vrouter running qos# policer pol1
vrouter running policer pol1#! bandwidth 1G
vrouter running policer pol1# burst 2K
vrouter running policer pol1# ..
vrouter running qos#
```

Create a shared policer that references the policer template:

```
vrouter running qos# shared-policer shared-pol1
vrouter running shared-policer shared-pol1# policer pol1
vrouter running shared-policer shared-pol1# ..
vrouter running qos#
```

Show the qos configuration:

```
vrouter running qos# show config
qos
  policer pol1
    bandwidth 1G
    burst 2K
    excess-bandwidth 0
    ..
  shared-policer shared-poll1
    policer pol1
    ..
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter running qos# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <qos xmlns="urn:6wind:vrouter/qos">
    <policer>
      <name>pol1</name>
      <burst>2000</burst>
      <excess-bandwidth>0</excess-bandwidth>
      <bandwidth>1000000000</bandwidth>
    </policer>
    <shared-policer>
      <name>shared-poll1</name>
      <policer>pol1</policer>
    </shared-policer>
  </qos>
</config>
```

---

**Note:** While the `policer` command defines traffic profile templates, that are instantiated whenever they are referenced, the `shared-policer` command defines unique objects.

---

### Rate limit an interface with a dedicated policer

Physical and logical interfaces can rate limit their ingress and egress traffic by attaching a dedicated policer, defined in the `qos` context.

Enter the `qos` context of physical interface `eth0`:

```
vrouter running config# vrf main
vrouter running vrf main# interface physical eth0
vrouter running physical eth0# qos
```

Configure rate limiting of egress traffic by policer poll:

```
vrouter running qos# egress rate-limit policer poll1
vrouter running qos# ..
vrouter running physical eth0#
```

Show interface eth0 configuration:

```
vrouter running physical eth0# show config nodefault
physical eth0
  (...)
  qos
    egress
      rate-limit
        policer poll1
      ..
    ..
  ..
```

Commit the configuration:

```
vrouter running physical eth0# commit
Configuration committed.
vrouter running physical eth0# /
vrouter running config#
```

Show interface qos state:

```
vrouter running config# show state vrf main interface
qos
  egress
    rate-limit
      policer
        bandwidth 1500M
        burst 1500
        excess-bandwidth 0
        excess-burst 1
        stats
          pass-packets 0
          pass-bytes 0
```

(continues on next page)

(continued from previous page)

```

                pass-excess-packets 0
                pass-excess-bytes 0
                drop-packets 0
                drop-bytes 0
                ..
            ..
        ..
    ..

```

The same settings can be made using the following NETCONF XML configuration:

```

vrouter running config# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <physical>
        <name>eth0</name>
        (...)
        <qos>
          <egress>
            <rate-limit>
              <policer>poll</policer>
            </rate-limit>
          </egress>
        </qos>
      </physical>
    </interface>
  </vrf>
</config>

```

Each interface that specifies `rate-limit policer poll` instantiates a new policer dedicated to the interface in the specified direction (ingress or egress).

### Rate limit interfaces with a shared policer

Physical and logical interfaces can rate limit their ingress and egress traffic by binding to a shared policer, defined in the qos context.

Enter the qos context of physical interface eth0:

```

vrouter running config# vrf main
vrouter running vrf main# interface physical eth0

```

(continues on next page)

(continued from previous page)

```
vrouter running physical eth0# qos
vrouter running qos#
```

Configure rate limiting of egress traffic by shared policer shared-pol1:

```
vrouter running qos# egress rate-limit shared-policer shared-pol1
vrouter running qos# ..
vrouter running physical eth1# ..
vrouter running interface#
```

Enter the qos context of physical interface eth1:

```
vrouter running interface# physical eth1
vrouter running physical eth1# qos
vrouter running qos#
```

Configure rate limiting of egress traffic by shared policer shared-pol1:

```
vrouter running qos# egress rate-limit shared-policer shared-pol1
vrouter running qos# ..
vrouter running physical eth1# ..
vrouter running interface#
```

Show interface eth0 configuration:

```
vrouter running interface# show config nodefault
interface
  physical eth0
    (...)
    qos
      egress
        rate-limit
          shared-policer shared-pol1
          ..
        ..
      ..
    ..
  physical eth1
    (...)
    qos
      egress
        rate-limit
          shared-policer shared-pol1
          ..
        ..
```

(continues on next page)

(continued from previous page)

```

..
..

```

Commit the configuration:

```

vrouters running interface# commit
Configuration committed.
vrouters running interface# /
vrouters running config#

```

Show interface qos state:

```

vrouters running config# show state vrf main interface
interface
  (...)
  physical eth0
    (...)
    qos
      egress
        rate-limit
          policer
            bandwidth 1G
            burst 2K
            excess-bandwidth 0
            excess-burst 1
            shared-policer shared-poll
            stats
              pass-packets 0
              pass-bytes 0
              pass-excess-packets 0
              pass-excess-bytes 0
              drop-packets 0
              drop-bytes 0
              ..
            ..
          ..
        ..
      ..
    ..
  physical eth1
    (...)
    qos
      egress
        rate-limit

```

(continues on next page)

(continued from previous page)

```

    policer
      bandwidth 1G
      burst 2K
      excess-bandwidth 0
      excess-burst 1
      shared-policer shared-poll
      stats
        pass-packets 0
        pass-bytes 0
        pass-excess-packets 0
        pass-excess-bytes 0
        drop-packets 0
        drop-bytes 0
        ..
      ..
    ..
  ..
..

```

The same settings can be made using the following NETCONF XML configuration:

```

<config xmlns="urn:6wind:vrouter">
  <qos xmlns="urn:6wind:vrouter/qos">
    <policer>
      <name>poll1</name>
      <burst>2000</burst>
      <excess-bandwidth>0</excess-bandwidth>
      <excess-burst>1</excess-burst>
      <bandwidth>1000000000</bandwidth>
    </policer>
    <shared-policer>
      <name>shared-poll1</name>
      <policer>poll1</policer>
    </shared-policer>
  </qos>
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <physical>
        <name>eth0</name>
        (...)
      <qos>
        <egress>

```

(continues on next page)

(continued from previous page)

```

        <rate-limit>
            <shared-policer>shared-pol1</shared-policer>
        </rate-limit>
    </egress>
</qos>
</physical>
<physical>
    <name>eth1</name>
    (...)
    <qos>
        <egress>
            <rate-limit>
                <shared-policer>shared-pol1</shared-policer>
            </rate-limit>
        </egress>
    </qos>
</physical>
</interface>
(...)
```

Each interface that specifies `rate-limit shared-policer pol1` uses the same shared policer object.

A given `shared-policer` may be shared by interfaces in different vrfs and directions.

**See also:**

The command reference for details on the qos global context:

- *qos context*

and for configuring qos on network interfaces:

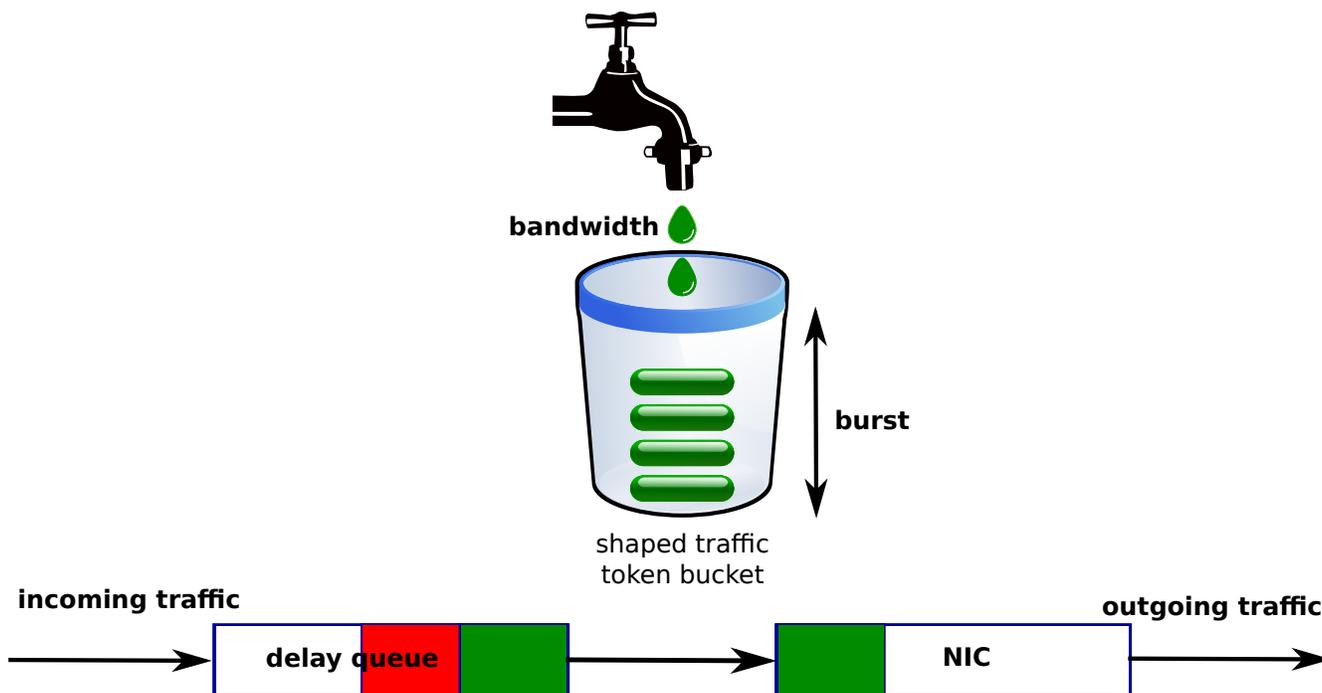
- *bridge interfaces qos*
- *gre interfaces qos*
- *ipip interfaces qos*
- *lag interfaces qos*
- *loopback interfaces qos*
- *physical interfaces qos*
- *veth interfaces qos*
- *vlan interfaces qos*
- *vxlan interfaces qos*
- *xvrf interfaces qos*

## Shaping

Shaping causes a traffic flow to conform to a bandwidth value referred to as the shaping rate. Excess traffic beyond the shaping rate is queued inside the shaper and transmitted only when doing so does not violate the defined shaping rate.

A shaper is implemented using a token bucket. If a packet fulfills the bandwidth/burst specification, it can pass. Otherwise, the packet is kept in a delay queue until it fulfills the bandwidth/burst specification. As soon as the delay queue is full, the incoming packets are dropped.

Shaping is applied to egress traffic on physical interfaces.



## Shaper templates

Shaper templates are created in the global qos context with the `shaper` command. They can then be referenced by a physical interface for egress.

Enter the global qos context and create a shaper:

```
vrouters running config# qos
vrouters running qos#
vrouters running qos# shaper shaper1
vrouters running shaper shaper1#! bandwidth 1G
vrouters running shaper shaper1# burst 2K
vrouters running shaper shaper1# queue-size 128
```

(continues on next page)

(continued from previous page)

```
vrouter running shaper shaper1# ..
vrouter running qos#
```

Interfaces that use this shaper will have their frame bandwidth shaped to 1 Gbps, with bursts up to 2 Kbytes. Frames that would cause this profile to be exceeded will be temporary saved in a delay queue to be sent later to fulfill the frame rate limitation. When the delay queue is full, the incoming frames are dropped.

By default the size of the delay queue is 256 packets. It can be changed via the `queue-size` command.

```
vrouter running config# qos
vrouter running qos#
vrouter running qos# shaper shaper1
vrouter running shaper shaper1# queue-size 128
vrouter running shaper shaper1# ..
vrouter running qos#
```

**Note:** If a scheduler and a shaper template are applied on an interface, the queue size of the shaper template is ignored. In this case the different queues of the scheduler are also used as delay queues.

In order to take into account bytes added to the frame size by the layer 1 level (by default 24 bytes for Ethernet CRC, Internet Frame Gap and preamble), you can specify an amount of bytes to add to the frame size in rate and burst calculations via the `layer1-overhead` command.

```
vrouter running config# qos
vrouter running qos#
vrouter running qos# shaper shaper2
vrouter running shaper shaper2#! bandwidth 10G
vrouter running shaper shaper2# layer1-overhead 24
vrouter running shaper shaper2# ..
vrouter running qos#
```

Review the QoS (Quality of Service) configuration:

```
vrouter running# show config qos
qos
  shaper shaper1
    bandwidth 1G
    burst 2K
    queue-size 128
    layer1-overhead 0
  ..
  shaper shaper2
    bandwidth 10G
```

(continues on next page)

(continued from previous page)

```

queue-size 256
layer1-overhead 24
..
..

```

The same settings can be applied using the following NETCONF XML configuration:

```

vrouters running config qos# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <qos xmlns="urn:6wind:vrouter/qos">
    <shaper>
      <name>shaper1</name>
      <burst>2000</burst>
      <layer1-overhead>0</layer1-overhead>
      <queue-size>128</queue-size>
      <bandwidth>10000000000</bandwidth>
    </shaper>
    <shaper>
      <name>shaper2</name>
      <layer1-overhead>24</layer1-overhead>
      <queue-size>256</queue-size>
      <bandwidth>10000000000</bandwidth>
    </shaper>
  </qos>
</config>

```

## Configuring a shaper on an interface

Shapers are configured in the qos context of physical interfaces.

Enter the qos context of the eth0 physical interface:

```

vrouters running config# vrf main
vrouters running vrf main# interface physical eth0
vrouters running physical eth0# qos

```

Configure shaper2 as the rate limiter for egress traffic:

```

vrouters running qos# egress rate-limit shaper shaper2
vrouters running qos# ..
vrouters running physical eth0#

```

Review eth0 configuration:

```

vrouters running physical eth0# show config nodefault
physical eth0
  (...)
  qos
    egress
      rate-limit
        shaper shaper1
      ..
    ..
  ..

```

Commit the configuration:

```

vrouters running physical eth0# commit
Configuration committed.
vrouters running physical eth0# /
vrouters running config#

```

Review the QoS state of the interface:

```

vrouters running config# show state vrf main interface physical eth0
physical eth0
  qos
    egress
      rate-limit
        shaper
          bandwidth 10G
          burst 125M
          queue-size 256
          layer1-overhead 24
          stats
            pass-packets 0
            drop-packets 0
          ..
        ..
      ..
    ..
  ..

```

The same settings can be applied using the following NETCONF XML configuration:

```

vrouters running config# show config xml absolute

```

(continues on next page)

(continued from previous page)

```
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <physical>
        <name>eth0</name>
        (...)
      </physical>
      <qos>
        <egress>
          <rate-limit>
            <shaper>shaper1</shaper>
          </rate-limit>
        </egress>
      </qos>
    </interface>
  </vrf>
</config>
```

## Protecting critical control plane traffic

When configuring a simple shaper on the output of an interface that is constantly fed with traffic over the limit, a part of the traffic is necessarily dropped. There is a risk that critical control plane traffic be dropped.

In order to protect this critical control plane and preserve it from being dropped, a simple QoS scheduler and traffic filter should be configured in addition to the shaper. Please refer to chapter *Shaping the output while protecting critical control plane traffic*.

## Scheduling

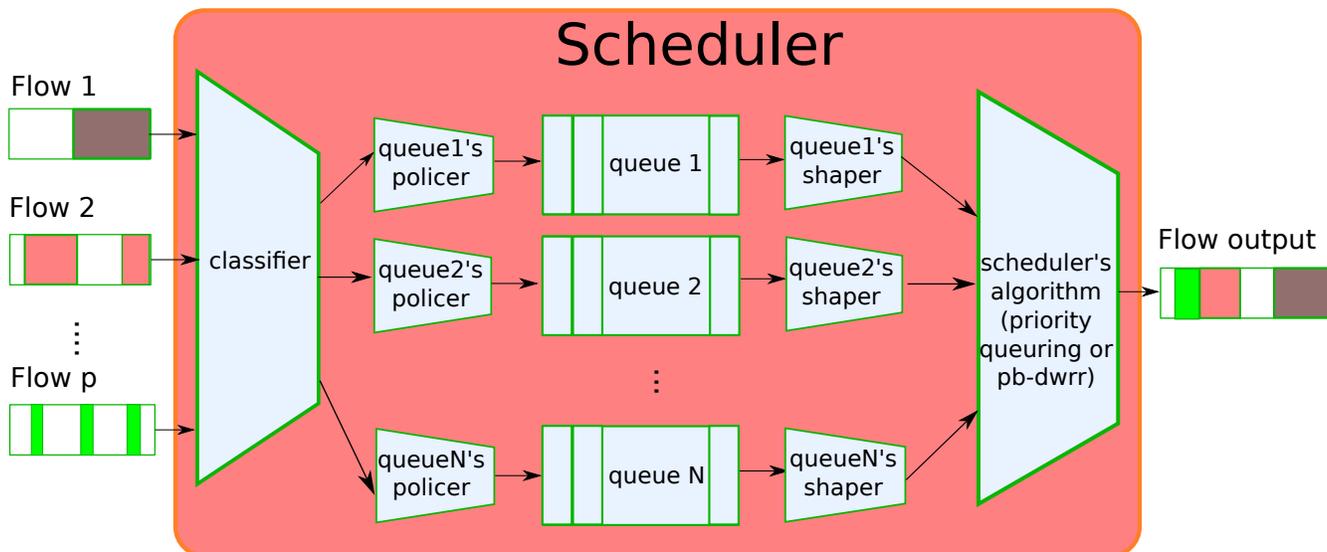
Scheduling allows to apply different types of processing to different egress queues configured on an interface. It assumes that the traffic is mapped to each queue, thanks to the concept of traffic class.

Scheduling provides two different queueing processings: Priority Queueing and PB-DWRR (Priority-Based Deficit Weighted Round Robin).

Each queue has several parameters:

- The size of the queue defines how many packets can be stored in the queue. Longer queues mean longer delays.
- The list of traffic classes that are submitted to the queue. Traffic classes are defined in the firewall section.
- An input policer to rate limit incoming traffic.

- An output shaper to rate limit outgoing traffic.
- Specific parameters related to the selected queuing processing (Priority Queuing or PB-DWRR)



Scheduling is applied to egress traffic on physical interfaces.

### Traffic classes

A class specifies a set of traffic flows, identified by their mark and/or the value of the CP\_FLAG (critical Control Plane traffic flag).

A class is attached to the queue in which traffic flows will be scheduled. One or more classes may be attached to the same queue.

### The critical control plane traffic flag

The CP\_FLAG is a flag set on critical packets sent by the control plane, i.e. the same traffic as the one protected by the *Control Plane Protection* feature, as described in the *control plane protection*.

This flag is automatically set when the Linux kernel outputs such a packet. It can be matched by the QoS classification stage.

## Flow matching and marking

Classes are defined by the mark of the packets and/or the value of the CP `FLAG`. Packet marking is done before QoS processing and must be configured at the *IP packet filtering* level.

QoS classification is usually based on the DSCP field of the IP header. In practice, this field is set for incoming packets by the border routers of a QoS network, allowing core routers to work with it without giving up too much processing power.

This example shows how to mark packets based on DSCP field:

```
vrouter running config# / vrf main
vrouter running vrf main# firewall ipv4
vrouter running ipv4# mangle
vrouter running mangle# prerouting
vrouter running prerouting# rule 1000 dscp af41 inbound-interface eth1 action mark 0x23
```

This example shows how to mark packets based on DSCP field, and change the DSCP value. Only one action is allowed per rule, therefore this requires to either use a chain or repeat the same rule with different actions.

```
vrouter running config# / vrf main
vrouter running vrf main# firewall ipv4
vrouter running ipv4# mangle
vrouter running mangle# chain g3 rule 1 action mark 0x3
vrouter running mangle# chain g3 rule 2 action dscp af41
vrouter running mangle# chain g3 policy return
vrouter running mangle# prerouting
vrouter running prerouting# rule 3000 dscp af42 inbound-interface eth1 action chain g3
```

Here, the chain policy is `return`, which means that matching packets will be processed by the remaining rules after being marked. Use policy `accept` if marked packets should not be processed further by the firewall.

---

**Note:** Refer to the `mark` action of the `rule` command in the *command reference*.

---

## Classification

Classes are created in the global `qos` context with the `class` command. They can then be referenced by any scheduler.

A class is defined by a packet mark and/or the value of the CP `FLAG`.

Enter the global `qos` context and create classes:

```
vrouter running config# / qos
vrouter running qos# class voip
```

(continues on next page)

(continued from previous page)

```
vrouters running class voip#! mark 0x1
vrouters running class voip# ..
vrouters running qos# class mail
vrouters running class mail#! mark 0x2
vrouters running class mail# ..
vrouters running qos#
```

By default, all bits of the mark are used to specify classes. Therefore, up to  $2^{32}$  different marks are supported. It is possible to specify which bits are used for QoS in order to use the mark for different purposes. In this case, the number of different marks is  $2^n$  where  $n$  is the number of bits reserved for the QoS in the mark.

To modify the mask used by the QoS enter the global qos context and edit the class-mask:

```
vrouters running config# / qos
vrouters running qos# class-mask 0xff
```

With this configuration, the first 8 bits of the mark are used to specify classes for QoS, so that 256 different marks can be used.

---

**Note:** The class-mask bitmask indicates which bits of the packet and class marks are taken into account. Other bits are ignored.

A packet belongs to a class if:

```
(packet-mark XOR class-mark) AND class-mask = 0
```

For example, with the following configuration:

```
/ qos class-mask 0xff
/ qos class class42 mark 0x42
/ qos class class542 mark 0x542
```

the 2 classes class42 and class542 match the same packets, those with a mark whose last byte is 0x42; for example packets with marks 0x42, 0x542, 0xff42 or 0x424242. Which class these packets will be assigned is undefined.

Therefore, care must be taken to avoid defining two classes for which (class-mark AND class-mask) equals the same value.

---

A class can also be configured to match (or to not match) the output critical control plane traffic.

The following class matches all critical control plane traffic:

```
vrouters running config# / qos
vrouters running qos# class control
vrouters running class control#! cp true
```

(continues on next page)

(continued from previous page)

```
vrouter running class control# ..  
vrouter running qos#
```

The following class matches all critical control plane traffic with mark 0x30:

```
vrouter running qos# / qos  
vrouter running qos# class control30  
vrouter running class control30#! cp true  
vrouter running class control30# mark 0x30  
vrouter running class control30# ..  
vrouter running qos#
```

The following class matches traffic with mark 0x30, except critical control plane traffic:

```
vrouter running qos# / qos  
vrouter running qos# class nocontrol30  
vrouter running class nocontrol30#! cp false  
vrouter running class nocontrol30#! mark 0x30  
vrouter running class nocontrol30# ..
```

---

**Note:** At most one mark and the value of the CP FLAG may be specified in a class. A packet belongs to a class if it matches all parameters specified in the class. The following combinations are supported, and evaluated in this order:

- cp true + mark
  - cp true
  - cp false + mark
  - mark
- 

**Note:** A packet that does not belong to any class or whose class is not bound to any queue will be submitted to the last queue.

---

## Scheduling algorithms

### Priority Queueing

When the scheduling algorithm is Priority Queueing, N queues are defined. Each queue has a different priority. The first queue has the highest priority, the last one has the lowest. Queues are served by order of priority: the scheduler first takes packets from the highest priority queue and submits them to the network hardware. When the queue is empty, it starts processing the next queue and so on.

### PB-DWRR

When the scheduling algorithm is PB-DWRR, N queues and two priority levels are defined: high and low.

Among the N queues, one has the high priority, and the N-1 others the low priority. Each low priority queue has a quantum that defines the share of the remaining bandwidth it will receive.

The high priority queue is served first. Once it is empty, other queues are served in a round robin fashion: the scheduler performs DWRR rounds between low priority queues. At each round, it checks each queue in sequence and enables it to send bytes up to its quantum. Then it serves the next queue, and so on.

When queue priorities are not set, all queues are served according to their quantum. This is the simple DWRR mode, which prevents starvation.

### Scheduler templates

Scheduler templates are created in the global qos context with the scheduler command. They can then be referenced by a physical interface for egress.

Enter the global qos context and create a scheduler using Priority Queueing:

```
vrouter running config# / qos
vrouter running qos# scheduler sched1
vrouter running scheduler sched1#! pq
vrouter running pq#! nb-queue 3
vrouter running pq# queue 1
vrouter running queue 1# class control
vrouter running queue 1# class voip
vrouter running queue 1# shaper shaper1
vrouter running queue 1# ..
vrouter running pq# queue 2
vrouter running queue 2# class mail
vrouter running queue 2# .. .. ..
```

Enter the global qos context and create a scheduler using PB-DWRR:

```
vrouter running config# / qos
vrouter running qos# scheduler sched2
vrouter running scheduler sched2#! core 2
vrouter running scheduler sched2#! pb-dwrr
vrouter running pb-dwrr#! nb-queue 3
vrouter running pb-dwrr# queue 1
vrouter running queue 1# class control
vrouter running queue 1# class voip
vrouter running queue 1# shaper shaper1
vrouter running queue 1# priority high
vrouter running queue 1# ..
vrouter running pb-dwrr# queue 2
vrouter running queue 2# class mail
vrouter running queue 2# quantum 3000
vrouter running queue 2# .. .. .
```

---

**Note:** A scheduler runs on a dedicated core which is chosen automatically if not set.

---

---

**Note:** To send several types of traffic flows to the same queue, you can define a class for each traffic flow, and attach all classes to the queue. (e.g. classes `control` and `voip` attached to queue 1 in examples above).

---

Review the QoS configuration:

```
vrouter running config# / qos
vrouter running qos# show config
qos
  shaper shaper1
    bandwidth 1G
    burst 2K
    layer1-overhead 0
    queue-size 128
    ..
  shaper shaper2
    bandwidth 10G
    burst 48K
    layer1-overhead 24
    queue-size 256
    ..
  scheduler sched1
    pq
      nb-queue 3
      queue 1
```

(continues on next page)

(continued from previous page)

```
        size 256
        shaper shaper1
        class control
        class voip
        ..
    queue 2
        size 256
        class mail
        ..
    ..
..
scheduler sched2
  core 2
  pb-dwrr
    nb-queue 3
    queue 1
        size 256
        shaper shaper1
        class control
        class voip
        quantum 1500
        priority high
        ..
    queue 2
        size 256
        class mail
        quantum 3000
        priority low
        ..
    ..
..
class-mask 0xff
class voip
  mark 0x1
  ..
class mail
  mark 0x2
  ..
class control
  cp true
  ..
class control30
  mark 0x30
  cp true
```

(continues on next page)

(continued from previous page)

```

..
class nocontrol30
  mark 0x30
  cp false
..
..

```

The same settings can be applied using the following NETCONF XML configuration:

```

vrouters running config# / qos
vrouters running config qos# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <qos xmlns="urn:6wind:vrouter/qos">
    <class-mask>0xff</class-mask>
    <class>
      <name>voip</name>
      <mark>0x1</mark>
    </class>
    <class>
      <name>mail</name>
      <mark>0x2</mark>
    </class>
    <class>
      <name>control</name>
      <cp>true</cp>
    </class>
    <class>
      <name>control30</name>
      <cp>true</cp>
      <mark>0x30</mark>
    </class>
    <class>
      <name>nocontrol30</name>
      <cp>>false</cp>
      <mark>0x30</mark>
    </class>
  <shaper>
    <name>shaper1</name>
    <burst>2000</burst>
    <layer1-overhead>0</layer1-overhead>
    <queue-size>128</queue-size>
    <bandwidth>1000000000</bandwidth>
  </shaper>
</qos>
</config>

```

(continues on next page)

(continued from previous page)

```
<name>shaper2</name>
<burst>48000</burst>
<layer1-overhead>24</layer1-overhead>
<queue-size>256</queue-size>
<bandwidth>10000000000</bandwidth>
</shaper>
<scheduler>
  <name>sched1</name>
  <pq>
    <nb-queue>3</nb-queue>
    <queue>
      <id>1</id>
      <size>256</size>
      <class>
        <name>mail</name>
      </class>
    </queue>
  </pq>
</scheduler>
<scheduler>
  <name>sched2</name>
  <core>2</core>
  <pb-dwrr>
    <nb-queue>3</nb-queue>
    <queue>
      <id>1</id>
      <size>256</size>
      <quantum>1500</quantum>
      <priority>high</priority>
      <class>
        <name>control</name>
      </class>
      <class>
        <name>voip</name>
      </class>
      <shaper>shaper1</shaper>
    </queue>
    <queue>
      <id>2</id>
      <size>256</size>
      <quantum>3000</quantum>
      <priority>low</priority>
      <class>
        <name>mail</name>
      </class>
    </queue>
  </pb-dwrr>
</scheduler>
```

(continues on next page)

(continued from previous page)

```
</class>
</queue>
</pb-dwrr>
</scheduler>
</qos>
</config>
```

## Configuring a scheduler on an interface

Schedulers are configured in the qos context of physical interfaces.

Enter the qos context of the eth0 physical interface:

```
vrouter running config# vrf main
vrouter running vrf main# interface physical eth0
vrouter running physical eth0# qos
```

Configure sched1 as the scheduler for egress traffic:

```
vrouter running qos# egress scheduler sched1
vrouter running qos# egress rate-limit shaper shaper2
vrouter running qos# ..
vrouter running physical eth0#
```

---

**Note:** When a scheduler is configured on an interface, it is mandatory to also configure a rate limit shaper on the same interface.

---

Review eth0 configuration:

```
vrouter running physical eth0# show config nodefault
physical eth0
  (...)
  qos
    egress
      rate-limit
        shaper shaper2
      ..
      scheduler sched1
    ..
  ..
..
```

Commit the configuration:

```

vrouter running physical eth0# commit
Configuration committed.
vrouter running physical eth0# ..
vrouter running config#

```

Review the QoS state of the interface:

```

qos
  egress
    rate-limit
      shaper
        bandwidth 10G
        burst 48K
        layer1-overhead 24
        queue-size 256
        stats
          pass-packets 0
          drop-packets 0
          ..
        ..
      ..
    scheduler
      core 2
      pq
        nb-queue 3
        queue 1
          size 256
          shaper
            bandwidth 1G
            burst 2K
            stats
              pass-packets 0
              drop-packets 0
              ..
            ..
          class cp
            stats
              match-packets 0
              ..
            ..
          class 0x00000001
            stats
              match-packets 0
              ..
            ..
          ..

```

(continues on next page)

(continued from previous page)

```

        stats
            enqueue-packets 0
            xmit-packets 0
            drop-queue-full 0
            ..
        ..
    queue 2
        size 256
        class 0x00000002
        stats
            match-packets 0
            ..
        ..
        stats
            enqueue-packets 0
            xmit-packets 0
            drop-queue-full 0
            ..
        ..
    queue 3
        size 256
        class default
        stats
            match-packets 0
            ..
        ..
        stats
            enqueue-packets 0
            xmit-packets 0
            drop-queue-full 0
            ..
        ..
    ..
    ..
    ..

```

The same settings can be applied using the following NETCONF XML configuration:

```

vrouter running config# show config xml absolute vrf main interface physical eth0
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">

```

(continues on next page)

(continued from previous page)

```

<physical>
  <name>eth0</name>
  (...)
  <qos>
    <egress>
      <scheduler>sched1</scheduler>
      <rate-limit>
        <shaper>shaper2</shaper>
      </rate-limit>
    </egress>
  </qos>
</physical>
</interface>
</vrf>
</config>

```

### Shaping the output while protecting critical control plane traffic

When configuring a simple shaper on the output of an interface that is constantly fed with traffic over the limit, a part of the traffic is necessarily dropped. There is a risk that critical control plane traffic be dropped.

A simple solution is to configure a Priority Queueing scheduler with 2 queues, one for the critical control plane traffic, the other for the rest of the traffic.

In this example, we configure an output shaper at 100Mbps, and the 2 queue Priority Queueing scheduler:

Configure the shaper wanshaper:

```

vrouter running config# / qos
vrouter running qos# shaper wanshaper
vrouter running shaper wanshaper#! bandwidth 100M
vrouter running shaper wanshaper# burst 125K
vrouter running shaper wanshaper# ..
vrouter running qos#

```

Configure the class for critical control plane traffic control:

```

vrouter running qos# class control
vrouter running class control#! cp true
vrouter running class control# ..
vrouter running qos#

```

Configure the scheduler wansched, and attach class control to the high priority queue:

```

vrouters running qos# scheduler wansched
vrouters running scheduler wansched#! pq
vrouters running pq#! nb-queue 2
vrouters running pq# queue 1
vrouters running queue 1# class control
vrouters running queue 1# ..
vrouters running pq# ..
vrouters running scheduler wansched# ..
vrouters running qos#

```

Attach the shaper and scheduler to the output interface:

```

vrouters running qos# / vrf main
vrouters running vrf main# interface physical eth0
vrouters running physical eth0# qos
vrouters running qos# egress
vrouters running egress# rate-limit
vrouters running rate-limit# shaper wanshaper
vrouters running rate-limit# ..
vrouters running egress# scheduler wansched
vrouters running egress# ..
vrouters running qos# ..
vrouters running physical eth0# ..

```

Review eth0 configuration:

```

vrouters running physical eth0# show config nodefault
physical eth0
  (...)
  qos
    egress
      rate-limit
        shaper wanshaper
        ..
      scheduler wansched
    ..
  ..
..

```

Commit the configuration:

```

vrouters running physical eth0# commit
Configuration committed.
vrouters running physical eth0# /
vrouters running config#

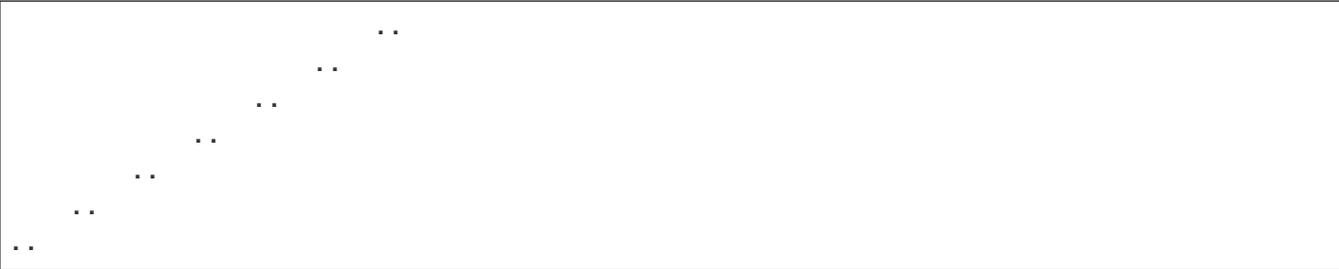
```

Review the QoS state of the interface:

```
vrouter running config# show state vrf main interface physical eth0
qos
  egress
    rate-limit
      shaper
        bandwidth 1M
        burst 1250
        layer1-overhead 0
        queue-size 256
        stats
          pass-packets 311640
          drop-packets 329125
          ..
        ..
      ..
    scheduler
      core 1
        pq
          nb-queue 2
          queue 1
            size 256
            class cp
            stats
              match-packets 90
              ..
            ..
          stats
            enqueue-packets 90
            xmit-packets 90
            drop-queue-full 0
            ..
          ..
        queue 2
          size 256
          class default
          stats
            match-packets 640583
            ..
          ..
        stats
          enqueue-packets 311550
          xmit-packets 311550
          drop-queue-full 329125
```

(continues on next page)

(continued from previous page)



The same settings can be applied using the following NETCONF XML configuration:

```
<config xmlns="urn:6wind:vrouter">
  (...)
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <physical>
        <name>eth0</name>
        <port>pci-b0s5</port>
        <qos>
          <egress>
            <rate-limit>
              <shaper>wanshaper</shaper>
            </rate-limit>
            <scheduler>wansched</scheduler>
          </egress>
        </qos>
      </physical>
    </vrf>
  <qos xmlns="urn:6wind:vrouter/qos">
    <class>
      <name>control</name>
      <cp>true</cp>
    </class>
    <shaper>
      <name>wanshaper</name>
      <burst>125000</burst>
      <bandwidth>100000000</bandwidth>
    </shaper>
    <scheduler>
      <name>wansched</name>
      <pq>
        <nb-queue>2</nb-queue>
      </pq>
      <queue>
        <id>1</id>
      </queue>
    </scheduler>
  </qos>
</config>
```

(continues on next page)

(continued from previous page)

```
<class>
  <name>control</name>
</class>
</queue>
</pq>
</scheduler>
</qos>
</config>
```

### 3.1.9 Security

---

**Note:** IKE requires a Turbo IPsec Application License.

---

#### IKE

Internet Key Exchange (IKE) is the control plane protocol providing authentication and key exchange mechanisms to establish secure VPNs over IPsec.

IKE peers authenticate each other via native IKE methods (pre-shared keys or certificates), or via various EAP (Extensible Authentication Protocol) methods.

#### About IPsec

IPsec (Internet Protocol Security) is a suite of protocols that provides security to Internet communications at the IP layer. The most common current use of IPsec is to provide a Virtual Private Network (VPN), either between two locations (gateway-to-gateway) or between a remote user and an enterprise network (host-to-gateway). More information is available in RFC4301.

#### About IKE

IKE (Internet Key Exchange) is the key negotiation and management protocol that is most commonly used to provide dynamically negotiated and updated keying material for IPsec. IPsec and IKE can be used in conjunction with both IPv4 and IPv6.

More information is available in RFC2409 and the latest update RFC7296.

The following sections explain the basics of IKE configuration, then present a couple of use cases and finally detail advanced configuration and performance tuning.

- *IKE configuration overview*
  - *Enabling IKE*
  - *VPN templates*
  - *IKE policy templates*
  - *IPsec policy templates*
  - *Creating a VPN*
- *IKE authentication*
  - *Pre-shared key authentication*
  - *Certificate authentication*
  - *EAP authentication*
- *IKE state*
- *Route-based VPNs*
  - *Static SVTI interfaces*
  - *Dynamic SVTI interfaces*
  - *Cross-VRF with static SVTI interfaces*
  - *Cross-VRF with dynamic SVTI interfaces*
- *Use cases*
  - *Use case: site to site VPN*
  - *Use case: VPN concentrator*
  - *Use case: route-based VPN concentrator*
- *Advanced configuration, performance and scalability*
  - *Logging*
  - *Extended Sequence Number (ESN)*
  - *Replay window size*
  - *Virtual IP pools*
  - *Retransmission constants*
  - *Lifetime of SA acquire messages*
  - *DoS protection*
  - *IKE worker threads*
  - *IKE SA hash table parameters*

- *IPsec SP hash table parameters*
- *Reverse route injection*
- *IKEv2 Mobility and Multihoming Protocol (MOBIKE)*

## IKE configuration overview

### Enabling IKE

IKE is enabled per VRF as follows:

```
vrouter running config# vrf main
vrouter running vrf main# ike
vrouter running ike#
```

Next, a VPN must be defined to specify the security parameters and policies to apply to the traffic, as well as authentication credentials for the IKE negotiation. To simplify the configuration of VPNs, VPN templates are proposed.

### VPN templates

The number of parameters for IKE is very high and it would be painful to repeat all of them for each VPN configuration. Therefore a template system is available to ease the configuration:

- several VPNs can share the same settings by referring to the same template,
- each parameter present in a template can be overridden by the VPN.

The IKE protocol consists of two phases:

- The first phase performs mutual authentication of two IKE peers and establishes an IKE Security Association (IKE SA), i.e. a secure communication channel between the two parties.
- The second phase enables to create or update pairs of ESP or AH SAs. Each pair of ESP or AH SAs is called a CHILD SA.

### IKE policy templates

IKE policy templates enable to define a model of IKE SA parameters. VPNs inherit their IKE SA parameters from such template, then can override each of them.

Create an IKE policy template:

```
vrouter running ike# ike-policy-template iketempl
vrouter running ike-policy-template iketempl#
```

The IKE policy template is initialized with various default values:

```
vrouter running ike-policy-template iketempl# show config
ike-policy-template iketempl
  local-auth-method pre-shared-key
  remote-auth-method pre-shared-key
  keying-tries 1
  unique-sa no
  reauth-time 0s
  rekey-time 4h
  dpd-delay 0s
  aggressive false
  udp-encap false
  ..
```

One or more IKE cryptographic algorithm proposals may then be defined in the `ike-policy-template`, or directly in the VPN `ike-policy`:

Each IKE proposal must contain either:

- a list of encryption algorithms (`enc-alg`).
- a list of authentication algorithms (`auth-alg`).
- a list of diffie hellman groups (`dh-group`) for key exchanges.
- optionally a list of pseudo-random function algorithms (`prf-alg`). If no `prf-alg` is provided, then the authentication algorithms will be used for generating random numbers.

Or:

- a list of combined mode algorithms (`aead-alg`), which provide both encryption and authentication.
- a list of diffie hellman groups (`dh-group`) for key exchanges.
- a list of pseudo-random function algorithms (`prf-alg`) for generating random numbers.

```
vrouter running ike-policy-template iketempl# ike-proposal 1
vrouter running ike-proposal 1#! enc-alg aes128-cbc
vrouter running ike-proposal 1#! auth-alg hmac-sha512
vrouter running ike-proposal 1#! dh-group modp2048
vrouter running ike-proposal 1# ..
vrouter running ike-policy-template iketempl# ..
vrouter running ike#
```

```
vrouter running ike# show config nodelist
ike
  (...)
  ike-policy-template iketempl
    ike-proposal 1
```

(continues on next page)

(continued from previous page)

```

    enc-alg aes128-cbc
    auth-alg hmac-sha512
    dh-group modp2048
    ..
  ..
..

```

As supported by the IKE protocol, the IKE daemon may submit several IKE proposals in a negotiation, and (for IKEv2 only), each proposal may contain several algorithms of the same type (for example several encryption algorithms).

All other parameters of an `ike-policy-template` have a default value. Each parameter (including `ike-proposal`) may be overridden by the VPN, for example the authentication method.

### IPsec policy templates

IPsec policy templates enable to define a model of CHILD SA parameters. VPNs inherit their IPsec SA parameters from such template, then can override each of them.

Create an IPsec policy template:

```

vrouter running ike# ipsec-policy-template ipsectemp1
vrouter running ipsec-policy-template ipsectemp1#

```

The IPsec policy template is initialized with various default values:

```

vrouter running ipsec-policy-template ipsectemp1# show config
ipsec-policy-template ipsectemp1
  start-action trap
  close-action trap
  dpd-action restart
  replay-window 32
  rekey-time 1h
  rekey-bytes 0
  rekey-packets 0
  encap-copy-dscp true
  decap-copy-dscp false
  encap-copy-df true
  ..

```

One or more ESP and AH cryptographic algorithm proposals may then be defined in the `ipsec-policy-template`, or directly in the VPN `ipsec-policy`.

Each ESP proposal must contain either:

- a list of encryption algorithms (`enc-alg`).

- a list of authentication algorithms (auth-`alg`).

Or:

- a list of combined mode algorithms (aead-`alg`), which provide both encryption and authentication.

```
vrouter running ike# ipsec-policy-template ipsectemp1
vrouter running ipsec-policy-template ipsectemp1# esp-proposal 1
vrouter running esp-proposal 1#! enc-alg aes128-cbc
vrouter running esp-proposal 1#! auth-alg hmac-sha256
vrouter running esp-proposal 1# ..
vrouter running ipsec-policy-template ipsectemp1# ..
vrouter running ike#
```

```
vrouter running ike# show config nodelist
ike
  (...)
  ipsec-policy-template ipsectemp1
    esp-proposal 1
      enc-alg aes128-cbc
      auth-alg hmac-sha256
    ..
  ..
..
```

Each AH proposal must contain:

- a list of authentication algorithms (auth-`alg`).

```
vrouter running ike# ipsec-policy-template ipsectemp1
vrouter running ipsec-policy-template ipsectemp1# ah-proposal 1
vrouter running ah-proposal 1#! auth-alg hmac-sha512
vrouter running ah-proposal 1# ..
vrouter running ipsec-policy-template ipsectemp1# ..
vrouter running ike#
```

```
vrouter running ike# show config nodelist
ike
  (...)
  ipsec-policy-template ipsectemp1
    (...)
    ah-proposal 1
      auth-alg hmac-sha512
    ..
  ..
..
```

Each ESP and AH proposal may optionally activate Perfect Forward Secrecy (PFS) by specifying a list of diffie hellman groups. This will trigger an additional diffie hellman exchange to exchange CHILD SA keys. If no `dh-group` is specified, CHILD SA keys will be derived from former keys.

```
vrouter running ike# ipsec-policy-template ipsectemp1
vrouter running ipsec-policy-template ipsectemp1# esp-proposal 1
vrouter running esp-proposal 1# dh-group modp2048
vrouter running esp-proposal 1# ..
vrouter running ipsec-policy-template ipsectemp1# ..
vrouter running ike#
```

```
vrouter running ike# show config nodefault
ike
  (...)
  ipsec-policy-template ipsectemp1
    (...)
    esp-proposal 1
      enc-alg aes128-cbc
      auth-alg hmac-sha256
      dh-group modp2048
      ..
    ..
  ..
```

A proposal may also optionally enable Extended Sequence Numbers (ESN) (see *Extended Sequence Number (ESN)*).

As supported by the IKE protocol, the IKE daemon may submit several ESP or AH proposals in a negotiation, and (for IKEv2 only), each proposal may contain several algorithms of the same type (for example several encryption algorithms).

All other parameters of an `ipsec-policy-template` have a default value. Each parameter (including `esp-proposal` and `ah-proposal`) may be overridden by the VPN, for example the replay window size.

An important parameter is `start-action` that defaults to `trap`, meaning that the tunnel will be triggered as soon as outgoing matching traffic is detected.

**See also:**

The *command reference* for details about template parameters.

To display the configuration, from the `ike` context, type:

```
vrouter running ike# show config
ike
  (...)
  ike-policy-template iketemp1
    local-auth-method pre-shared-key
    remote-auth-method pre-shared-key
```

(continues on next page)

(continued from previous page)

```
keying-tries 1
reauth-time 0s
rekey-time 4h
dpd-delay 0s
aggressive false
udp-encap false
ike-proposal 1
    enc-alg aes128-cbc
    auth-alg hmac-sha256
    dh-group modp2048
    auth-alg hmac-sha512
    ..
..
ipsec-policy-template ipsectemp1
start-action trap
close-action trap
dpd-action restart
replay-window 32
rekey-time 1h
rekey-bytes 0
rekey-packets 0
encap-copy-dscp true
decap-copy-dscp false
encap-copy-df true
esp-proposal 1
    enc-alg aes128-cbc
    auth-alg hmac-sha256
    ..
ah-proposal 1
    auth-alg hmac-sha512
    ..
..
```

After VPN templates have been created, you may use them in one or several VPNs.

## Creating a VPN

A VPN defines the security parameters between the local host and a remote IKE peer (or a group of IKE peers), and the IPsec security policies to apply to the IP traffic that transits through these peers.

Creating a VPN basically consists in:

- specifying which IKE and IPsec template to apply,
- optionally overriding some parameters of these templates,

- define identities of the peers and their credentials,
- specify the IPsec security policies to apply.

Create the vpn *vpn-hq*, use the ike-policy-template *iketempl* and override parameter *keying-tries*, use the ipsec-policy-template *ipsectempl*.

```
vrouter running vpn vpn-hq#! ike-policy
vrouter running ike-policy#! template iketempl
vrouter running ike-policy#! keying-tries 10
vrouter running ike-policy#! ..
vrouter running vpn vpn-hq#! ipsec-policy
vrouter running ipsec-policy#! template ipsectempl
vrouter running ipsec-policy#! ..
vrouter running vpn vpn-hq#! local-address 192.0.2.1
vrouter running vpn vpn-hq#! remote-address 198.51.100.1
vrouter running vpn vpn-hq#! local-id user1.roadw.6wind.net
vrouter running vpn vpn-hq#! remote-id secgw.6wind.net
```

Then define an IPsec security-policy *trunk* between subnets 192.168.0.0/24 and 192.168.99.0/24, with the default action (do ESP in tunnel mode).

```
vrouter running vpn vpn-hq#! security-policy trunk
vrouter running security-policy trunk#! local-ts subnet 192.168.0.0/24
vrouter running security-policy trunk#! remote-ts subnet 192.168.99.0/24
vrouter running security-policy trunk#! ..
vrouter running vpn vpn-hq#! ..
vrouter running ike#
```

```
vrouter running ike# show config nodefault
ike
  (...)
  ike-policy-template iketempl
    ike-proposal 1
      enc-alg aes128-cbc
      auth-alg hmac-sha512
      dh-group modp2048
      ..
    ..
  ipsec-policy-template ipsectempl
    esp-proposal 1
      enc-alg aes128-cbc
      auth-alg hmac-sha256
      ..
    ..
  vpn vpn-hq
```

(continues on next page)

(continued from previous page)

```

ike-policy
  template iketemp1
  keying-tries 10
  ..
ipsec-policy
  template ipsectemp1
  ..
local-address 192.0.2.1
remote-address 198.51.100.1
local-id user1.roadw.6wind.net
remote-id secgw.6wind.net
security-policy trunk
  local-ts subnet 192.168.0.0/24
  remote-ts subnet 192.168.99.0/24
  ..
..
..

```

Finally, define a pre-shared key *hq-secgw* for mutual authentication with the remote peer:

```

vrouter running ike# pre-shared-key hq-secgw
vrouter running pre-shared-key hq-secgw#! id 198.51.100.1
vrouter running pre-shared-key hq-secgw#! secret 0seaJ31RfzHNRvUSH0oUYg7znTW0I=
vrouter running pre-shared-key hq-secgw# ..
vrouter running ike#

```

```

vrouter running ike# show config nodefaut
ike
  pre-shared-key hq-secgw
  id 198.51.100.1
  secret 0seaJ31RfzHNRvUSH0oUYg7znTW0I=
  ..
global-options
  dos-protection
  ..
  sp-hash-ipv4
  sp-hash-ipv6
  ..
ike-policy-template iketemp1
  ike-proposal 1
  enc-alg aes128-cbc
  auth-alg hmac-sha512
  dh-group modp2048
  ..

```

(continues on next page)

(continued from previous page)

```
..
ipsec-policy-template ipsectemp1
  esp-proposal 1
    enc-alg aes128-cbc
    auth-alg hmac-sha256
    ..
  ..
vpn vpn-hq
  ike-policy
    template iketemp1
    keying-tries 10
    ..
  ipsec-policy
    template ipsectemp1
    ..
  local-address 192.0.2.1
  remote-address 198.51.100.1
  local-id user1.roadw.6wind.net
  remote-id secgw.6wind.net
  security-policy trunk
    local-ts subnet 192.168.0.0/24
    remote-ts subnet 192.168.99.0/24
    ..
  ..
..
```

## IKE authentication

Configuring IKE authentication consists in:

- choosing the local and remote authentication methods (pre-shared keys, certificate signatures or an EAP method),
- specifying the local (and optionally remote) authentication identity,
- configuring keys, certificates or contact information of a RADIUS (Remote Authentication Dial-In User Service) server.

The authentication methods of the local and remote IKE peer may be asymmetric: For example, the local host may authenticate by certificate and the remote peer by EAP.

The methods used to authenticate the local and remote peer are specified in the `ike-policy-template` and may be overridden in the VPN `ike-policy`:

```
vrouters running ike# vpn vpn-hq
vrouters running vpn vpn-hq# ike-policy
vrouters running ike-policy# local-auth-method certificate
vrouters running ike-policy# remote-auth-method eap-mschapv2
vrouters running ike-policy# ..
vrouters running vpn vpn-hq#
```

If unspecified, the default authentication method is pre-shared-key.

The local IKE identity is defined in the VPN:

```
vrouters running vpn vpn-hq# local-id server@6wind.com
```

If unspecified, the local IKE identity defaults to:

- the peer IP address for pre-shared key
- the certificate subject for certificate authentication

When using certificate authentication, the IKE identity must be contained in the certificate, either as subject or as subjectAltName.

Optionally, the remote IKE identity may be specified. It indicates which identity to expect for the authentication round. It also enables to choose the right pre-shared key when initiating a negotiation.

If EAP authentication is used, the local or remote EAP identity is defined by a different command:

```
vrouters running vpn vpn-to-hq# local-eap-id client1@6wind.com
```

If unspecified, the EAP identity defaults to the IKE identity.

If the remote EAP identity is set to %any, the client will be asked for its EAP identity via the EAP-Identity method.

```
vrouters running vpn vpn-hq# remote-eap-id %any
```

## Pre-shared key authentication

Pre-shared keys are secret symmetric keys shared by two IKE peers. They are configured in the pre-shared-key list.

When using pre-shared key authentication for the local host or remote peer authentication, the shared key must be declared as follows:

```
vrouters running ike# pre-shared-key hq-secgw
vrouters running pre-shared-key hq-secgw#! id 198.51.100.1
vrouters running pre-shared-key hq-secgw#! secret 0seaJ31RfzHNRvUSH0oUYg7znTW0I=
vrouters running pre-shared-key hq-secgw# ..
vrouters running ike#
```

```
vrouters running ike# show config
ike
  (...)
  pre-shared-key hq-secgw
    id secgw.6wind.net
    secret 0seaJ31RfzHNRvUSH0oUYg7znTW0I=
  ..
```

Each pre-shared key is identified by this name and is composed of two parts, a secret key and optional IKE identity selectors (a list of IKE identities).

The secret key itself, `secret`, may be encoded either:

- as a sequence of characters delimited by double-quotes,

```
secret "this is a weak password"
```

- as an hexadecimal binary value, prefixed by `0x`:

```
secret 0xd2c79a277d517f31cd46f5121f4a14620ef39d35b4
```

- a base64 binary value, prefixed by `0s`:

```
secret 0seaJ31RfzHNRvUSH0oUYg7znTW0I=
```

The optional IKE identity selector list `id`, specifies for which peers this key must be used. To authenticate a connection between two hosts, the entry that most specifically matches the host and peer IDs is used.

An entry with a single selector matches if the peer ID matches the selector. An entry with multiple selectors matches if both the local host ID and peer ID each match one of the selectors. An entry with no ID matches all peers, it is the default pre-shared key.

For more information, see [strongSwan's IKE secrets ID selectors](https://wiki.strongswan.org/projects/strongswan/wiki/IpsecSecretors) (<https://wiki.strongswan.org/projects/strongswan/wiki/IpsecSecretors>).

To authenticate the local host by pre-shared keys, the `local-auth-method` must be set to `pre-shared-key` in the `ike-policy-template` used by the VPN, or overridden in the VPN `ike-policy`.

```
vrouters running ike# ike-policy-template ikepsk local-auth-method pre-shared-key
vrouters running ike# vpn vpn-hq ike-policy template ikepsk
```

or:

```
vrouters running ike# vpn vpn-hq ike-policy local-auth-method pre-shared-key
```

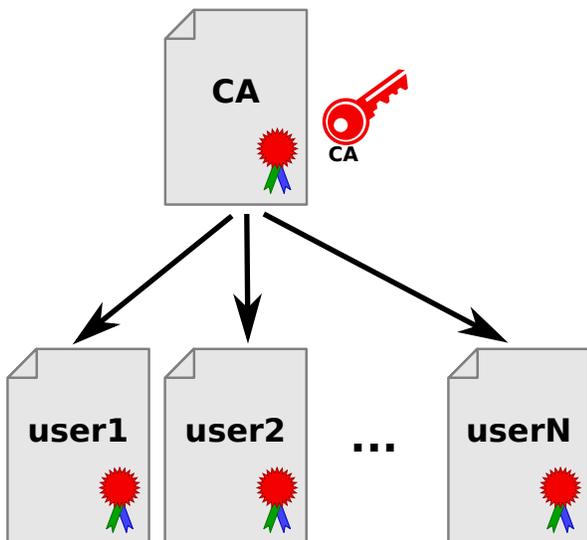
Similarly, to authenticate the remote peer by pre-shared keys, the `remote-auth-method` must be set to `pre-shared-key` in the `ike-policy-template` used by the VPN, or overridden in the VPN `ike-policy`.

Pre-shared keys is the default authentication method.

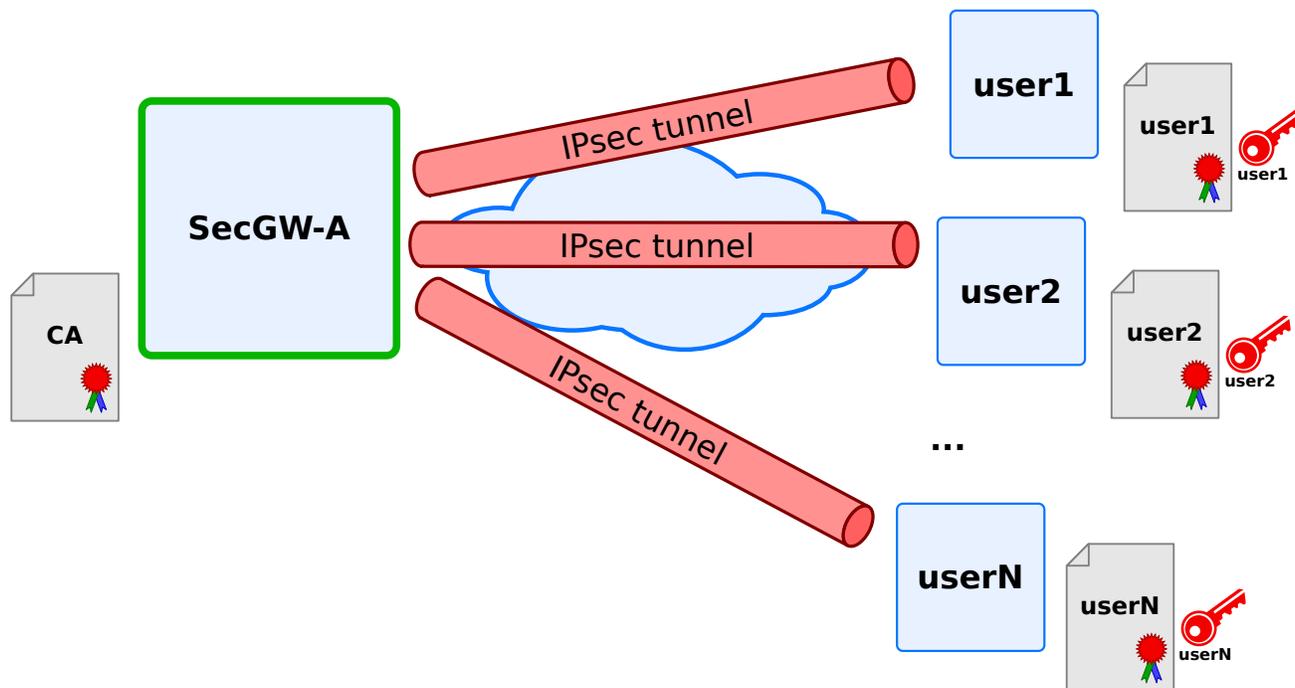
### Certificate authentication

Certificate authentication performs authentication via RSA public key cryptography.

Contrarily to pre-shared keys, certificates do not imply that the IKE peers exchange secret keys beforehand. To authenticate remote peers, an IKE endpoint simply needs to trust the certificate authority who delivered and signed the remote peers' certificates.



Certificates enable to easily deploy a large number of IKE clients without maintaining and distributing a large list of secret keys (one for each pair of IKE peers) or weakening the system by using a single secret key shared between all IKE peers. It also avoids to modify the configuration of each peer when a new one is added.



Each IKE peer owns a digital certificate and a private key. The certificate embeds identity information and the matching public key. The certificate is delivered and signed by a certificate authority (CA), whose public key is stored in a CA (Certification Authority) certificate. The CA certificate enables to validate the authenticity of all certificates that it delivered.

Like for bank cards, CAs (Certification Authorities) may also revoke a valid certificate before its expiration, for example in case of disclosure of the public key or the departure of an employee. To proceed, the CA may deliver a signed certificate revocation list (CRL), that lists revoked certificates.

Certificates, private keys and certificate revocation lists are stored in the Privacy Enhanced Mail (PEM) format in the configuration.

### Local host authentication by certificate

The local host certificate and private key must be installed in the certificate list:

```

vrouter running ike# certificate secgw-a
vrouter running certificate secgw-a#! certificate "-----BEGIN CERTIFICATE-----
... MIIB9jCCA V8CAQMwDQYJKoZIhvcNAQEEBQA wUzETMBEGA1UEChMKNldJTkQgUy5B
... LjeOMAwGA1UEBxMFUGFyaXMxCzAJBgNVBAYTAkZSMR8wHQYDVQDE xZIZWFkcXVh
... cnRlcnMgQXV0aG9yaXR5MB4XDTE4MDkxOTEzMjM1MloXDTE5MDkxOTEzMjM1Mlow
... NDELMAkGA1UEBhMCR1IxEzARBgNVBAoTCjZXSU5EIFMuQS4xEDA0BgNVBAMTB1Nl
... Y0dXLUeWgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBA0uCfHphepTnl1pX/emq
... IMjW35RAm3TSSHSgDvBm/QtBHgJgLd53ANGbRQ7o1inx7jA+CrbrBM9BdEXdR7So
... Q9++munDep/Eb9vu55mMm/1eZ8xnV4jIDjLmHCP/AMPNYzKVJHPCElDIbLsbvHIq

```

(continues on next page)

(continued from previous page)

```

... 8A6CYaQ0i7NkOrkRY9q3LiEzAgMBAAEwDQYJKoZIhvcNAQEEBQADgYEAAdSmmAN5+
... eRh7WuxuAlSGJh1PwB3NzrSKcbJnMPMz1qCqVhvQiGTQNIe5rpr6AlJN7LZV/wvS
... ng4yIizgehU0fluNfAroTEOoxq06m39YZPoY6mUNk82kRq3YTeX+j9EizRjePHzk
... jfYhCQITZa0atkjpfI143b0/k1NVC9exBv0=
... -----END CERTIFICATE-----"
vrouters running certificate secgw-a#! private-key "-----BEGIN PRIVATE KEY-----
... MIICdgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBAOuCFhphepTnl1pX
... /emqIMjW35RAm3TSSHSgDvBm/QtBHgJgLD53ANGbRQ7olinx7jA+CrbrBM9BdEXd
... R7SoQ9++munDep/Eb9vu55mMm/leZ8xnV4jIDjLmHCP/AMPNYzKVJHPCElDIbLsb
... vHIq8A6CYaQ0i7NkOrkRY9q3LiEzAgMBAAECYB7IBoiBUqIBNeXXf9ypS5Esgnr
... wSdFGRcmWfPVfZJ3ytB8n3n7n62+5/VfyPuQ7FoBwL3rSc2W6Xp3eCuf6ISquXy8
... zNIB2EY4dzXWpza9E8+0nZi08dzFyphM0BFN44pwSazrgD0ZSnXQbxzFBwm5+VvC
... cxSpR/A+53bxDklAIQJBAPnMBvgHdtZATV4rzUN42l//McSGgba1GklICul5rIk/
... GhkGLVLgRaxsJoM3myV7lwA/7jJwXX3ypnJE02uODXECQQDxW6JTUK5N2/0idS1i
... +Y/cEhgv0c7e3zTvTK3qe5t6Q1A2+1n6mpjk4iRSAfsiemudnUFIBqbCpyZ1/GeV
... 2JbJkAPau1fL67BCJT94/w2VuY7mJesxpSI/2KQ9VZfFLh2fCOTodNgUyFzXa8Y
... eD0mMhue01NTX6YVmp12/gkg2VKxAKAUMkLHdf1H7pykAYImwNTqv/zIG9bHvpi
... +9uhv24nMPLJZwcEfwNF49Z+NkQ5eYZQThRkXoodx7bkMJbKZfZAKEA+R+jxmK/
... /XiiT7zizYaWW5x/PQRGvpf0ehmlcp11+u03ILDo1NqD7gde98P9Rlc2xXF++K8I
... 3yyFFRutrqwKjw==
... -----END PRIVATE KEY-----"
vrouters running certificate secgw-a# ..
vrouters running ike#

```

```

vrouters running ike# show config nodefault
ike
(...)
certificate secgw-a
certificate "-----BEGIN CERTIFICATE-----
MIIB9jCCA8CAQMwDQYJKoZIhvcNAQEEBQAwUzETMBEGA1UEChMKNldJTkQgUy5B
LjEOMAwGA1UEBxMFUGFyaXMxZCZAJBgNVBAYTAkZSMR8wHQYDVQDEExZIZWFkcXVh
cnRlcnMgQXV0aG9yaXR5MjB4MDkxOTEzZmM1MloXDTE5MDkxOTEzZmM1Mlow
NDELMAkGA1UEBhMCRlIxEzARBgNVBAoTCjZXSU5EIFMuQS4xEDA0BgNVBAMTB1Nl
Y0dXLUeWgZ8wDQYJKoZIhvcNAQEEBQADgY0AMIGJAoGBAOuCFhphepTnl1pX/emq
IMjW35RAm3TSSHSgDvBm/QtBHgJgLD53ANGbRQ7olinx7jA+CrbrBM9BdEXdR7So
Q9++munDep/Eb9vu55mMm/leZ8xnV4jIDjLmHCP/AMPNYzKVJHPCElDIbLsbvHIq
8A6CYaQ0i7NkOrkRY9q3LiEzAgMBAAEwDQYJKoZIhvcNAQEEBQADgYEAAdSmmAN5+
eRh7WuxuAlSGJh1PwB3NzrSKcbJnMPMz1qCqVhvQiGTQNIe5rpr6AlJN7LZV/wvS
ng4yIizgehU0fluNfAroTEOoxq06m39YZPoY6mUNk82kRq3YTeX+j9EizRjePHzk
jfYhCQITZa0atkjpfI143b0/k1NVC9exBv0=
-----END CERTIFICATE-----"
private-key "-----BEGIN PRIVATE KEY-----
MIICdgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBAOuCFhphepTnl1pX

```

(continues on next page)

(continued from previous page)

```

/emqIMjW35RAm3TSSHSgDvBm/QtBHgJgLd53ANGbRQ7oLinX7jA+CrbrBM9BdEXd
R7SoQ9++munDep/Eb9vu55mMm/1eZ8xnV4jIDjLmHCP/AMPNYzKVJHPCElDIbLsb
vHIq8A6CYaQ0i7NkOrkRY9q3LiEzAgMBAAEcGyB7IBoiBUqIBNeXXf9ypS5Esgnr
wSdFGRcmWfPvfZJ3ytB8n3n7n62+5/VfyPuQ7FoBwL3rSc2W6Xp3eCuf6ISquXy8
zNIB2EY4dzXWpza9E8+0nZi08dzFyphM0BFN44pwSazrgD0ZSnXQbxzFBwm5+VvC
cxSpR/A+53bxDklAIQJBAPnMBvgHdtZATV4rzUN421//McSGgba1GklICul5rIk/
GhkGLVLgRaxsJoM3myV71wA/7jJwXX3ypnJEO2uODXECQDxW6JTUK5N2/0idSii
+Y/cEhgv0c7e3zTvTK3qe5t6Q1A2+1n6mpjk4iRSAfsiEMudnUFIBqbCpyZ1/GeV
2JbjAkAPau1fL67BCJT94/w2VuY7mJesxpSI/2KQ9VZfFLh2fCOT0dNgUyFZxA8Y
eD0mMhue01NTX6YVmP12/gkg2VKxAKaAUMkLHdf1H7pykAYImwhNTqv/zIG9bHvpi
+9uhv24nMPLJZwcEfwNF49Z+NkQ5eYZQThRkXoodx7bkMJbKZzFZAkEA+R+jxmK/
/XiIT7zizYaWW5x/PQRgvpfOehmlcp11+u03ILDolNqD7gde98P9Rlc2xXF++K8I
3yyFFRutrqwKjw==
-----END PRIVATE KEY-----"
..

```

Then the `local-auth-method` must be set to `certificate` in the `ike-policy-template` used by the VPN (or overridden in the VPN `ike-policy`).

Finally, the list of certificate candidates to use for authentication is specified in the VPN `certificate` command. The certificate used for authentication is selected based on the received certificate request payloads. If no appropriate CA can be located, the first certificate is used.

The IKE id used by the local host must be stored in its certificate, in the `subjectName` or in the `subjectAltNames` section.

```

vrouters running ike# vpn siteA-roadw
vrouters running vpn siteA-roadw#! ike-policy
vrouters running ike-policy#! template iketempl
vrouters running ike-policy#! local-auth-method certificate
vrouters running ike-policy#! ..
vrouters running vpn siteA-roadw#! ipsec-policy template ipsectempl
vrouters running vpn siteA-roadw# certificate secgw-a
vrouters running vpn siteA-roadw# ..
vrouters running ike#

```

```

vrouters running ike# show config
ike
  (...)
  vpn siteA-roadw
    ike-policy
      template iketempl
      local-auth-method certificate
    ..
    ipsec-policy

```

(continues on next page)

(continued from previous page)

```

    template ipsectemp1
    ..
    certificate secgw-a
    ..
    ..

```

## Remote peer authentication by certificate

The certificate authority that issued the certificates that remote peers will present must be declared in the certificate-authority list:

```

vrouters running ike# certificate-authority hq-authority
vrouters running certificate-authority hq-authority# certificate "-----BEGIN_
↪CERTIFICATE-----
... MIIC2zCCAkSgAwIBAgIJAjPUB7T8zBYBMA0GCSqGSIb3DQEBAUAMFMxEzARBgNV
... BAoTCjZXSU5EIFMuQS4xDjAMBgNVBAcTBVBhcm1zMQswCQYDVQGEWJGUjEfmB0G
... A1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhvcml0eTAeFw0xODA5MTkxMzE5MTNaFw0x
... ODEwMTkxMzE5MTNaMFMxEzARBgNVBAoTCjZXSU5EIFMuQS4xDjAMBgNVBAcTBVBh
... cm1zMQswCQYDVQGEWJGUjEfmB0GA1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhvcml0
... eTCBnzANBkqkhiG9w0BAQEFAAOBjQAwgYkCgYEA2mWsQQ14SSkx0Qp5eXXHMkAV
... OEyIJVD3dVPrcQkeCUR38KPrA8Dmlt/KLTrTfat6+/wxS1HywCLYR3U1+CrEQmR+
... kC/NgcNC+QqXyevb+2LTT606oHMq6XckWIDhhD6JszN0dtcAci1SMgaKIoaoxElu
... TwIdDBkj8W7gnpn84k8CAwEAAaOBtjCBszAMBgNVHRMEBTADAQH/MB0GA1UdDgQW
... BBSN5H+zxbYDk/kVJuqimYsT2oDGDTCBgwYDVR0jBHwweoAUjeR/s8W2A5P5FSbq
... opmLE9qAxc2hV6RVMFMxEzARBgNVBAoTCjZXSU5EIFMuQS4xDjAMBgNVBAcTBVBh
... cm1zMQswCQYDVQGEWJGUjEfmB0GA1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhvcml0
... eYIJAjPUB7T8zBYBMA0GCSqGSIb3DQEBAUAA4GBAEvu9Rj1dUcQsFywiseZdZcC7
... 9jxhHtml1naxqDp/krPG/GJiSiCypQOGjbcX1Ra2NOtLU7DwZTKH3S3fw8TBIAen
... 7vbQFLUtzrZ07TW4wnmtBtGd7GVqAZVioUnklDVHhHL6hGy2DM+3e8+lptx8+tb6
... U/7s2V3Bm/HkQRq8+Gji
... -----END CERTIFICATE-----"
vrouters running certificate-authority hq-authority# ..
vrouters running ike#

```

```

vrouters running ike# show config nodelist
ike
  (...)
  certificate-authority hq-authority
    certificate "-----BEGIN CERTIFICATE-----
MIIC2zCCAkSgAwIBAgIJAjPUB7T8zBYBMA0GCSqGSIb3DQEBAUAMFMxEzARBgNV
BAoTCjZXSU5EIFMuQS4xDjAMBgNVBAcTBVBhcm1zMQswCQYDVQGEWJGUjEfmB0G
A1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhvcml0eTAeFw0xODA5MTkxMzE5MTNaFw0x

```

(continues on next page)

(continued from previous page)

```

ODEwMTkxMzE5MTNaMFMxEzARBgNVBAoTCjZXSU5EIFMuQS4xDjAMBgNVBAcTBVBh
cm1zMQswCQYDVQQGEWJGUjEjFMB0GA1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhvcml0
eTCBnzANBQkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA2mWsQQ14SSkx0Qp5eXXHMkAV
OEyIJVD3dVPrCQkeCUR38KPrA8Dmlt/KLTrTfat6+/wxS1HywCLYR3U1+CrEQmR+
kC/NgcNC+QqXyevb+2LTT606oHMq6XckWIDhhD6JszN0dtcAci1SMgaKIoaoxElu
TwIdDBkj8W7gnpn84k8CAwEAAaOBtjCBszAMBgNVHRMEBTADAQH/MB0GA1UdDgQW
BBSN5H+zxbYDk/kVJuqimYsT2oDGDTCBgwYDVR0jBHwweoAUjeR/s8W2A5P5FSbq
opmLE9qAxxg2hV6RVMFMxEzARBgNVBAoTCjZXSU5EIFMuQS4xDjAMBgNVBAcTBVBh
cm1zMQswCQYDVQQGEWJGUjEjFMB0GA1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhvcml0
eYIJAjUB7T8zBYBMA0GCSqGSIb3DQEBAUAA4GBAEvu9Rj1dUcQsFywseZdZcC7
9jxhHtml1naxqDp/krPG/GjSiCypQOGjbcXlRa2N0tLU7DwZTKH3S3fw8TBIAen
7vbQFLUtzrZ07TW4wnmtBtGd7GVqAZVioUnkldVHhHL6hGy2DM+3e8+lptx8+tb6
U/7s2V3Bm/HkQRq8+Gji
-----END CERTIFICATE-----"

```

```

vrouter running ike#

```

Then to authenticate the remote peer by certificates, the `remote-auth-method` must be set to `certificate` in the `ike-policy-template` used by the VPN (or overridden in the VPN `ike-policy`).

Finally, the CA certificates to trust for the authentication of the remote peer must be specified in the VPN `remote-ca-certificate` list.

The IKE id used by the remote peer must be stored in its certificate, in the `subjectName` or in the `subjectAltNames` section.

```

vrouter running ike# vpn siteA-roadw
vrouter running vpn siteA-roadw#! ike-policy
vrouter running ike-policy#! template iketempl
vrouter running ike-policy#! remote-auth-method certificate
vrouter running ike-policy#! ..
vrouter running vpn siteA-roadw#! ipsec-policy template ipsectempl
vrouter running vpn siteA-roadw# remote-ca-certificate hq-authority
vrouter running vpn siteA-roadw# ..
vrouter running ike#

```

```

vrouter running ike# show config
ike
  (...)
  vpn siteA-roadw
    ike-policy
      template iketempl
      remote-auth-method certificate
    ..
  ipsec-policy

```

(continues on next page)

(continued from previous page)

```

template ipsectemp1
..
remote-ca-certificate hq-authority
..
..

```

## Manage revocation of remote peer certificates

Using certificates usually implies to handle certificate revocations.

To manually add a CRL (Certificate Revocation List), in PEM (Privacy Enhanced Mail) format:

```

vrouter running ike# certificate-authority hq-authority
vrouter running certificate-authority hq-authority# crl "-----BEGIN X509 CRL-----
... MIIBYjCCATMCAQEWdQYJKoZIhvcNAQEEBQAwwUzETMBEGA1UEChMKNldJTkQgUy5B
... LjEOMAwGA1UEBxMFUGFyaXMxCzAJBgNVBAYTAkZSMR8wHQYDVQQDEZXIZWFkcXVh
... cnRlcnMgQXV0aG9yaXR5Fw0xODA5MTkxMzI2MTlaFw0xODEwMTkxMzI2MTlaMBQw
... EgIBARcnMTgwOTE5MTMyMzI2MTlaFw0xODEwMTkxMzI2MTlaMBQw
... FSbqopmLE9qAyg2hV6RVFMxEzARBgNVBAoTCjZXSU5EIFMuQS4xDjAMBGNVBAcT
... BVBhcnMzMQswCQYDVQQGEWJGUjEjFMB0GA1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhv
... cm10eYIJAjUB7T8zBYBMAoGA1UdFAQDAgEBMA0GCSqGSIb3DQEBAUAA4GBAAAtY
... 3gXNIMwMjH6rafv9wI5qrDCwOp7KNdcrZbNuV/RURJ9mle8EPJ01PJSnxPMuIuzX
... VGbgjRxagWAQLl1j4bkhHiqiezThi0D5xTsmmmXEZ52oK5GVDjElWU90ZeK1vssLL
... PK9DsxuURw0RP32iv6l68qwaPdI4tR0K8wcVXPn9
... -----END X509 CRL-----"
vrouter running certificate-authority hq-authority# ..
vrouter running ike#

```

```

vrouter running ike# show config nodefault
ike
(...)
certificate-authority hq-authority
certificate (...)
crl "-----BEGIN X509 CRL-----
MIIBYjCCATMCAQEWdQYJKoZIhvcNAQEEBQAwwUzETMBEGA1UEChMKNldJTkQgUy5B
LjEOMAwGA1UEBxMFUGFyaXMxCzAJBgNVBAYTAkZSMR8wHQYDVQQDEZXIZWFkcXVh
cnRlcnMgQXV0aG9yaXR5Fw0xODA5MTkxMzI2MTlaFw0xODEwMTkxMzI2MTlaMBQw
EgIBARcnMTgwOTE5MTMyMzI2MTlaFw0xODEwMTkxMzI2MTlaMBQw
FSbqopmLE9qAyg2hV6RVFMxEzARBgNVBAoTCjZXSU5EIFMuQS4xDjAMBGNVBAcT
BVBhcnMzMQswCQYDVQQGEWJGUjEjFMB0GA1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhv
cm10eYIJAjUB7T8zBYBMAoGA1UdFAQDAgEBMA0GCSqGSIb3DQEBAUAA4GBAAAtY
3gXNIMwMjH6rafv9wI5qrDCwOp7KNdcrZbNuV/RURJ9mle8EPJ01PJSnxPMuIuzX
VGbgjRxagWAQLl1j4bkhHiqiezThi0D5xTsmmmXEZ52oK5GVDjElWU90ZeK1vssLL

```

(continues on next page)

(continued from previous page)

```
PK9DsxuURw0RP32iv6l68qwaPdI4tR0K8wcVXPn9
-----END X509 CRL-----"
..
..
```

To add a CRL distribution point, specify the LDAP (Lightweight Directory Access Protocol) or HTTP (HyperText Transfer Protocol) URI (Uniform Resource Identifier). CRLs (Certificate Revocation Lists) must be encoded in DER (Distinguished Encoding Rules) binary format on the distribution server.

```
vrouter running ike# certificate-authority hq-authority
vrouter running certificate-authority hq-authority# crl-uri ldap://hq-authority.6wind.
↵net
vrouter running certificate-authority hq-authority# ..
vrouter running ike#
```

```
vrouter running ike# show config nodefault
ike
  (...)
  certificate-authority hq-authority
    certificate (...)
    crl (...)
    crl-uri ldap://hq-authority.6wind.net
  ..
..
```

## EAP authentication

EAP is typically used by a VPN concentrator accepting IKE connections, to authenticate remote clients via external methods (legacy methods such as EAP-MD5 (EAP - Message Digest 5) or EAP-MSCHAPv2 (EAP - Microsoft CHAP v2), mobile network methods such as EAP-SIM (EAP - Subscriber Identity Module) or EAP-AKA (EAP - Authentication and Key Agreement)...). The authentication methods are usually asymmetric: the server is authenticated by pre-shared keys or a certificate, and the clients by EAP.

### Local and remote peer EAP authentication

Local and remote EAP keys may be stored in a local database. They are similar to pre-shared keys, but are used by EAP authentication methods. They are configured in the `eap-key` list.

These keys are looked up to authenticate IKE peers if the `local-auth-method` or `remote-auth-method` is set to `eap-md5` or `eap-mschapv2`.

```
vrouter running ike# eap-key user1key
vrouter running eap-key user1key#! id user1@6wind.com
vrouter running pre-shared-key user1key#! secret EAPpassword1
vrouter running pre-shared-key user1key# ..
vrouter running ike#
```

```
vrouter running ike# show config
ike
  (...)
  eap-key user1key
    id user1@6wind.com
    secret EAPpassword1
  ..
```

Like pre-shared keys, EAP keys are assigned a name and are composed of two parts, a secret key and optional EAP identity selectors (a list of EAP identities).

The encodings and selection rules are the same as for pre-shared keys, except that the EAP ID is taken into account instead of the IKE ID.

To authenticate the local host by EAP keys, the `local-auth-method` must be set to the right EAP method `eap-mschapv2` or `eap-md5` in the `ike-policy-template` used by the VPN, or overridden in the VPN `ike-policy`.

```
vrouter running ike# ike-policy-template ikepsk local-auth-method eap-mschapv2
vrouter running ike# vpn vpn-hq ike-policy template ikepsk
```

or:

```
vrouter running ike# vpn vpn-hq ike-policy local-auth-method eap-mschapv2
```

Similarly, to authenticate the remote peer by pre-shared keys, the `remote-auth-method` must be set to `eap-mschapv2` or `eap-md5` in the `ike-policy-template` used by the VPN, or overridden in the VPN `ike-policy`.

## Remote peer authentication by EAP via RADIUS

On the server side, the EAP authentication of remote peers can be delegated to one or more RADIUS servers, the IKE daemon then acts a simple proxy.

This delegation of EAP authentication to RADIUS servers is configured by selecting `eap-radius` as the remote authentication method, and by declaring one or more EAP RADIUS servers in the `eap-radius` list.

Select `eap-radius` as the remote authentication method in the VPN IKE policy:

```

router-vm running ike# vpn mytunnel
router-vm running vpn mytunnel#! ike-policy
router-vm running ike-policy#! template basic_policy
router-vm running ike-policy#! remote-auth-method eap-radius
router-vm running ike-policy#! ..
router-vm running vpn mytunnel#! ..
router-vm running ike#!

```

Configure an EAP RADIUS server. The minimal parameters are the server IP address and an authentication secret.

```

router-vm running ike# eap-radius
router-vm running eap-radius# server server-tnr
router-vm running server server-tnr#! address 10.200.0.1
router-vm running server server-tnr#! secret testing123
router-vm running server server-tnr# ..
router-vm running eap-radius# ..

```

Show the EAP RADIUS server configuration:

```

router-vm running ike# show config eap-radius
eap-radius
  nas-identifier 6WINDvRouter
  auth-port 1812
  sockets 1
  retransmit-tries 4
  retransmit-timeout 2.0
  retransmit-base 1.4
  server server-tnr
    address 10.200.0.1
    secret testing123
  ..
..

```

## IKE state

Show the IKE state:

```

vrouter running config# vrf main
vrouter running vrf main# ike
vrouter running ike# show state
ike
  enabled true
  pre-shared-key psk-hq
  id 10.125.0.2

```

(continues on next page)

(continued from previous page)

```
id 10.125.0.1
secret "This is a strong password"
..
logging
  daemon
    default 0
    ..
  authpriv
    default disable
    ..
  ..
global-options
  dos-protection
    cookie-threshold 10
    block-threshold 5
    init-limit-half-open 0
    ..
  threads 16
  acquire-timeout 30
  sa-table-size 1
  sa-table-segments 1
  sp-hash-ipv4 local 32 remote 32
  sp-hash-ipv6 local 128 remote 128
  install-routes false
  routing-table 220
  routing-table-prio 220
  retransmit-tries 5
  retransmit-timeout 4.0
  retransmit-base 1.8
  delete-rekeyed false
  delete-rekeyed-delay 5
  make-before-break false
  snmp false
  mobike-prefer-best-path false
  ..
ha
  enabled false
  ..
vpn vpn-hq
  version 2
  local-address 10.125.0.1
  remote-address 10.125.0.2
  security-policy site2site
  local-ts subnet 10.100.0.0/24
```

(continues on next page)

(continued from previous page)

```
remote-ts subnet 10.200.0.0/24
action esp
mode tunnel
priority 0
..
ike-policy
  ike-proposal 1
    enc-alg aes128-cbc
    auth-alg hmac-sha1
    dh-group modp2048
    ..
  local-auth-method pre-shared-key
  remote-auth-method pre-shared-key
  keying-tries 1
  unique-sa no
  reauth-time 0
  rekey-time 14400
  dpd-delay 0s
  aggressive false
  udp-encap false
  mobike false
  ..
ipsec-policy
  esp-proposal 1
    enc-alg aes128-cbc
    auth-alg hmac-sha1
    dh-group modp2048
    ..
  start-action trap
  close-action trap
  dpd-action restart
  replay-window 32
  rekey-time 3600
  rekey-bytes 0
  rekey-packets 0
  encap-copy-dscp true
  decap-copy-dscp false
  encap-copy-df true
  ..
..
ike-sas
total 1
half-open 0
..
```

(continues on next page)

(continued from previous page)

```
task-processing
  worker-threads
    total 16
    idle 11
    critical 4
    high 0
    medium 1
    low 0
    ..
  task-queues
    critical 0
    high 0
    medium 0
    low 0
    scheduled 3
    ..
  ..
counters
  ike-rekey-init 0
  ike-rekey-resp 0
  child-rekey 0
  invalid 0
  invalid-spi 0
  ike-init-in-req 0
  ike-init-in-resp 1
  ike-init-out-req 1
  ike-init-out-resp 0
  ike-auth-in-req 0
  ike-auth-in-resp 1
  ike-auth-out-req 1
  ike-auth-out-resp 0
  create-child-in-req 0
  create-child-in-resp 0
  create-child-out-req 0
  create-child-out-resp 0
  info-in-req 0
  info-in-resp 0
  info-out-req 0
  info-out-resp 0
  ..
vpn-counters name vpn-hq
  ike-rekey-init 0
  ike-rekey-resp 0
  child-rekey 0
```

(continues on next page)

(continued from previous page)

```
invalid 0
invalid-spi 0
ike-init-in-req 0
ike-init-in-resp 1
ike-init-out-req 1
ike-init-out-resp 0
ike-auth-in-req 0
ike-auth-in-resp 1
ike-auth-out-req 1
ike-auth-out-resp 0
create-child-in-req 0
create-child-in-resp 0
create-child-out-req 0
create-child-out-resp 0
info-in-req 0
info-in-resp 0
info-out-req 0
info-out-resp 0
..
ike-sa unique-id 1
  name vpn-hq
  version 2
  state established
  local-address 10.125.0.1
  remote-address 10.125.0.2
  local-port 500
  remote-port 500
  initiator-spi 6e6228d1c13daaf1
  responder-spi b2f0a5217f09662a
  enc-alg aes128-cbc
  auth-alg hmac-sha1
  prf-alg hmac-sha1
  dh-group modp2048
  established-time 24
  rekey-time 14170
  reauth-time 45567
  udp-encap false
  mobike false
  child-sa unique-id 2
    name site2site
    state installed
    reqid 1
    protocol esp
    udp-encap false
```

(continues on next page)

(continued from previous page)

```
mobike false
spi-in c704d981
spi-out c3dd14b9
enc-alg aes128-cbc
auth-alg hmac-sha1
esn false
bytes-in 304
packets-in 2
bytes-out 168
packets-out 2
installed-time 24
rekey-time 3425
life-time 3936
local-ts
    subnet 10.100.0.0/24
    ..
remote-ts
    subnet 10.200.0.0/24
    ..
..
remote-port 500
initiator-spi 6e6228d1c13daaf1
responder-spi b2f0a5217f09662a
enc-alg aes128-cbc
auth-alg hmac-sha1
prf-alg hmac-sha1
dh-group modp2048
established-time 24
rekey-time 14170
reauth-time 45567
udp-encap false
mobike false
child-sa unique-id 2
    name site2site
    state installed
    reqid 1
    protocol esp
    udp-encap false
    mobike false
    spi-in c704d981
    spi-out c3dd14b9
    enc-alg aes128-cbc
    auth-alg hmac-sha1
    esn false
```

(continues on next page)

(continued from previous page)

```

bytes-in 304
packets-in 2
bytes-out 168
packets-out 2
installed-time 24
rekey-time 3425
life-time 3936
local-ts
  subnet 10.100.0.0/24
  ..
remote-ts
  subnet 10.200.0.0/24
  ..
..
..
..

```

The state dumps:

- the applied configuration,
- the number of negotiated IKE SAs (`ike-sas`),
- information about the IKE daemon internal tasks (`task-processing`),
- global IKEv2 message counters (`counters`),
- per VPN IKEv2 message counters (`vpn-counters`). Note that when the host is responder, some counters remain null because the IKE daemon cannot determine the involved VPN before the authentication is completed (`invalid`, `invalid-spi`, `ike-init-in-req`, `ike-init-out-resp...`),
- the negotiated IKE SAs and their child SAs (`ike-sa`).

---

**Note:** Child SAs traffic selector proposed by the remote peer can include unsupported stuff (like port range). In this case, the flag `unsupported` is set:

**local-ts** subnet 10.100.0.0/24 protocol 17 port 50000-54000 unsupported

---

## Route-based VPNs

Security policies can be bound to SVTI interfaces to configure route-based VPNs.

SVTI interfaces handle their own SPD and SAD.

Outgoing traffic routed through an SVTI interface is submitted to a security policy lookup against the SVTI interface's own SPD and, when a matching SP (Security Policy) is found, encrypted using an SA from its own SAD matching the SP, or dropped if no match was found.

Incoming IPsec-encrypted traffic is first decrypted with the right SA. If the SA is bound to an SVTI interface (via an svti-id), it is then submitted to a security policy check against the SVTI interface's own SPD. If the packet is granted access, the decrypted traffic is received via the SVTI interface.

## Static SVTI interfaces

To bind a security policy to an SVTI interface, specify the svti-id of the interface on inbound and outbound policies:

Create SVTI interface svti100:

```
vrouter running config# vrf main interface svti svti100
vrouter running svti svti100#! svti-id 100
vrouter running svti svti100# /
vrouter running config#
```

Create IKE VPN my\_vpn, with a security policy bound to svti100:

```
vrouter running config# vrf main ike vpn my_vpn
vrouter running vpn my_vpn#!
(...)
vrouter running vpn my_vpn# security-policy mytunnel
vrouter running security-policy mytunnel# svti-id-in 100
vrouter running security-policy mytunnel# svti-id-out 100
```

The security policy is bound to the SVTI interface with SVTI ID 100 and link VRFID main, namely svti100.

**Note:** The decision to bind to an SVTI interface is done per security-policy and per direction. The configuration may differ between two security-policies and between two directions of the same security-policy. For example:

```
vrouter running vpn my_vpn# security-policy mytunnel1 svti-id-in 100 svti-id-out 200
vrouter running vpn my_vpn# security-policy mytunnel2 svti-id-out 150
vrouter running vpn my_vpn# security-policy mytunnel3
```

See *SVTI* for details about creating SVTI interfaces.

## Dynamic SVTI interfaces

SVTI interfaces may be dynamically created and attached to IKE SAs as they are established.

To proceed, first create an SVTI template:

```
vrouter running config# / vrf main
vrouter running vrf main# interface svti-template svtitemp100
vrouter running svti-template svtitemp100# mtu 1300
vrouter running svti-template svtitemp100# /
vrouter running config#
```

Then create a VPN bound to this SVTI template:

```
vrouter running config# / vrf main ike vpn my_vpn
vrouter running vpn my_vpn#!
(...)
vrouter running vpn my_vpn# dynamic-svti
vrouter running dynamic-svti# svti-template svtitemp100
vrouter running dynamic-svti# ..
vrouter running vpn my_vpn#
```

---

**Note:** A separate SVTI interface is created for each established IKE SA spawn from this VPN definition. All its child SAs are bound to the SVTI, for both the inbound and outbound traffic.

---

Routes are automatically added or deleted via the dynamically created SVTI interface as child SAs are established or torn down. Which routes should be added is configurable in the SVTI template with the `install-routes-to` command.

Add routes to the IKE SA remote host virtual IPs, if any, else to the child SA remote traffic selector. This is the default behavior:

```
vrouter running config# / vrf main
vrouter running vrf main# interface svti-template svtitemp100
vrouter running svti-template svtitemp100#
vrouter running svti-template svtitemp100# install-routes-to vip-or-remote-ts
```

Add routes to the child SA remote traffic selector:

```
vrouter running svti-template svtitemp100# install-routes-to remote-ts
```

Add routes to the IKE SA remote host virtual IPs, if any:

```
vrouter running svti-template svtitemp100# install-routes-to vip
```

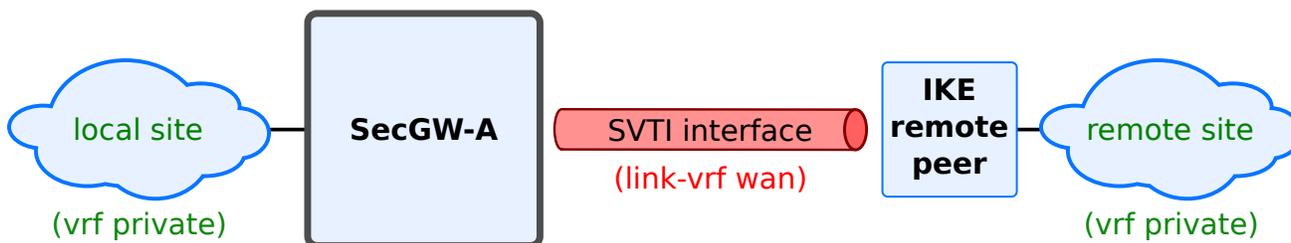
Do not add any route:

```
vrouter running svti-template svtitemp100# install-routes-to none
```

See *Dynamic SVTI* for details about creating SVTI interface templates.

### Cross-VRF with static SVTI interfaces

Like other tunnel interfaces, SVTI interfaces enable to perform cross-vrf encapsulation: encapsulated packets are not in the same vrf as the original packets.



To proceed, configure the SVTI interface in the VRF of original packets, specify a link VRF equal to the VRF of encapsulated packets.

```
vrouter running config# vrf private interface svti svti100
vrouter running svti svti100#! svti-id 100
vrouter running svti svti100# link-vrf wan
vrouter running svti svti100# /
vrouter running config#
```

Then configure IKE in the link VRF.

```
vrouter running config# vrf wan ike vpn my_vpn
vrouter running vpn my_vpn#!
(...)
vrouter running vpn my_vpn# security-policy mytunnel
vrouter running security-policy mytunnel# svti-id-in 100
vrouter running security-policy mytunnel# svti-id-out 100
vrouter running security-policy mytunnel# /
vrouter running config#
```

The SVTI interface is uniquely identified by its (svti-id, link-vrf) pair.

To show the configuration:

```
vrouter running config# show config nodelist
vrf private
  interface
    svti svti100
```

(continues on next page)

(continued from previous page)

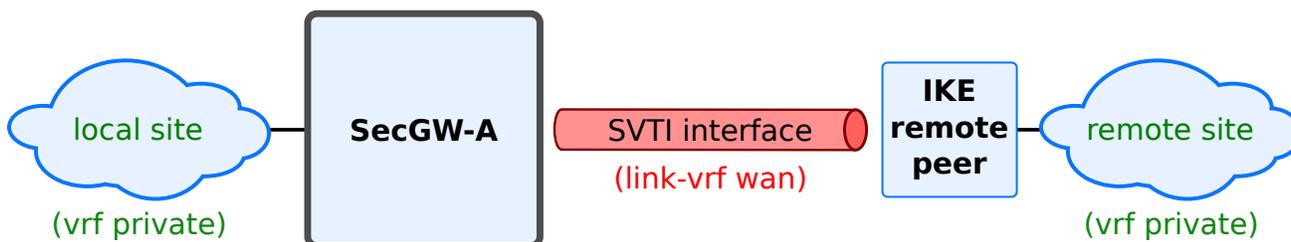
```

        link-vrf wan
        ..
        ..
    (...)
vrf wan
    ike
        vpn my_vpn
        (...)
        security-policy mytunnel
            svti-id-in 100
            svti-id-out 100
    (...)

```

### Cross-VRF with dynamic SVTI interfaces

Dynamic SVTI interfaces also enable to perform cross-vrf encapsulation. The link-vrf of a dynamic interface is the vrf of the VPN that triggered its creation.



To proceed, configure the SVTI template in the VRF of original packets.

```

vrouters running vrf private# interface svti-template svtitemp100
vrouters running svti-template svtitemp100# mtu 1300
vrouters running svti-template svtitemp100# /
vrouters running config#

```

Then create a VPN in the link VRF and bind it to the SVTI template, by specifying its name and VRF.

```

vrouters running config# vrf wan ike vpn my_vpn
vrouters running vpn my_vpn#!
(...)
vrouters running vpn my_vpn# dynamic-svti
vrouters running dynamic-svti# svti-template svtitemp100
vrouters running dynamic-svti# vrf private
vrouters running dynamic-svti# /
vrouters running config#

```

To show the configuration:

```
vrouter running config# show config nodefault
vrf private
  interface
    svti-template svtitemp100
    mtu 1300
    ..
  ..
  (...)
vrf wan
  ike
    vpn my_vpn
    dynamic-svti
      svti-template TEMP
      vrf private
    ..
  (...)
  (...)
```

---

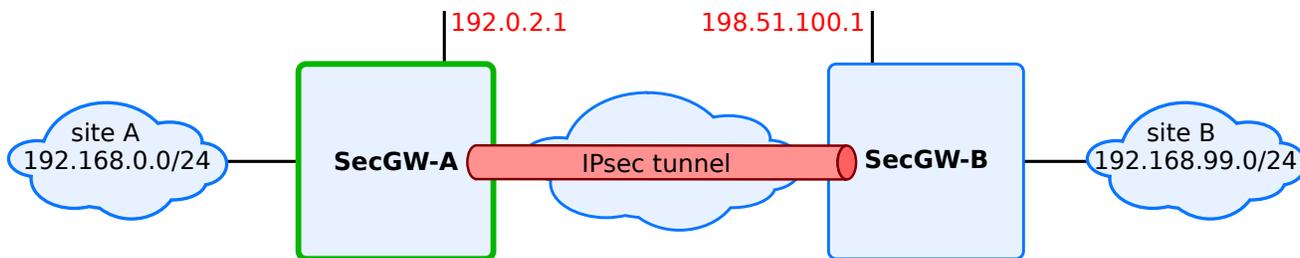
**Note:** If a VPN in VRF wan specifies an SVTI template, then no static SVTI must be configured with its link VRF in wan.

---

### Use cases

#### Use case: site to site VPN

In this use case, two sites A and B must be interconnected via a public network. An IPsec VPN is configured between the two security gateways SecGW-A and SecGW-B.

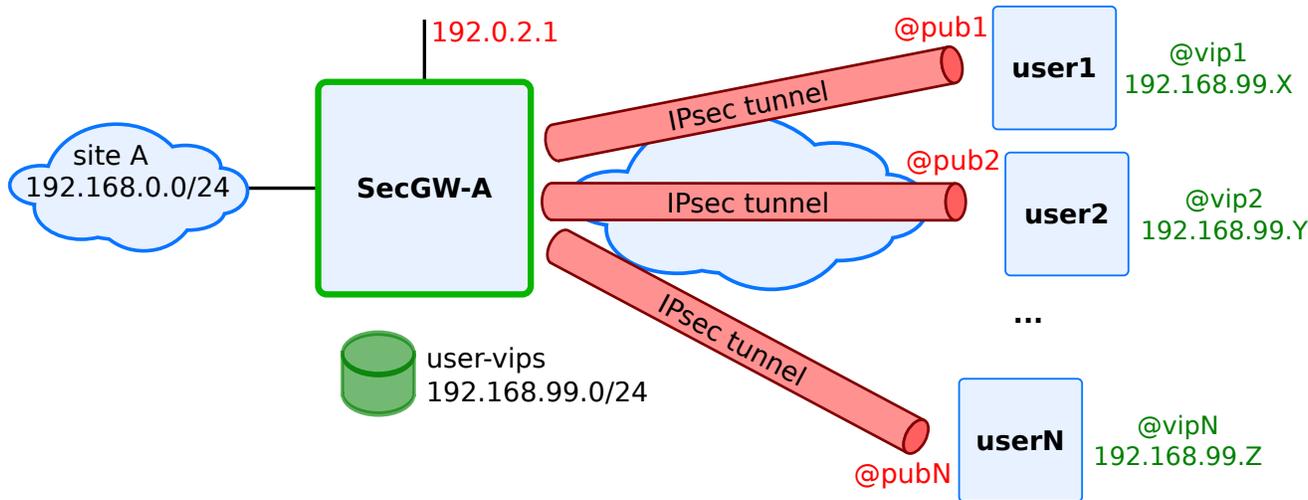


The IP addresses of the security gateways and of the sites are well known. The peers identify themselves with a Fully Qualified Domain Name (FQDN) and authenticate via a pre-shared key.

```
vrouter running ike# show config nodefault
ike
  global-options
  ..
  ike-policy-template iketemp1
    ike-proposal 1
      enc-alg aes128-cbc
      auth-alg hmac-sha512
      prf-alg hmac-sha512
      dh-group modp2048
      ..
    ..
  ipsec-policy-template ipsectemp1
    esp-proposal 1
      enc-alg aes128-cbc
      auth-alg hmac-sha256
      ..
    ah-proposal 1
      auth-alg hmac-sha512
      ..
    ..
  vpn siteA-siteB
    ike-policy
      template iketemp1
      ..
    ipsec-policy
      template ipsectemp1
      ..
    local-address 192.0.2.1
    remote-address 198.51.100.1
    local-id secgwa.6wind.net
    remote-id secgwb.6wind.net
    security-policy trunk
      local-ts subnet 192.168.0.0/24
      remote-ts subnet 192.168.99.0/24
      ..
    ..
  pre-shared-key siteb
    id secgwb.6wind.net
    secret 0seaJ31RfzHNRvUSH0oUYg7znTW0I=
    ..
```

### Use case: VPN concentrator

In this use case, remote users must be given access to the local site A via a public network. The traffic must be secured by IPsec VPNs between users and the security gateways SecGW-A.



IKE negotiations are initiated by the remote users. Their public IP addresses are dynamically assigned by their access point. Each user requests the security gateway to assign it a virtual private address. The security gateway picks this virtual IP from a local pool.

The peers identify themselves with a user Fully Qualified Domain Name (user FQDN) and authenticate via pre-shared keys. Remote hosts use different VPN clients that support different cryptographic algorithms and key lengths.

```
vrouter running ike# show config nodefault
ike
  global-options
  ..
  ike-policy-template iketemp1
    ike-proposal 1
      enc-alg aes256-cbc
      enc-alg aes128-cbc
      auth-alg hmac-sha512
      prf-alg hmac-sha512
      dh-group modp2048
      ..
    ike-proposal 2
      aead-alg aes128-gcm-128
      prf-alg hmac-sha512
      dh-group modp2048
      ..
  ..
```

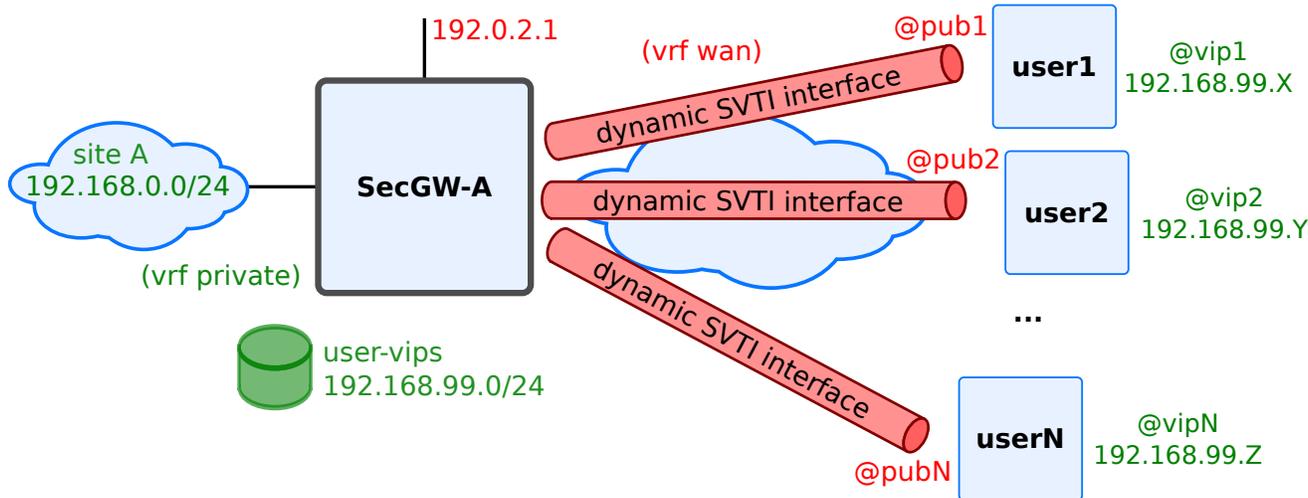
(continues on next page)

(continued from previous page)

```
ipsec-policy-template ipsectemp1
  esp-proposal 1
    enc-alg aes128-cbc
    auth-alg hmac-sha256
    ..
  esp-proposal 2
    aead-alg aes128-gcm-128
    ..
  ah-proposal 1
    auth-alg hmac-sha512
    ..
  ..
vpn siteA-roadw
  ike-policy
    template iketemp1
    ..
  ipsec-policy
    template ipsectemp1
    ..
  local-address 192.0.2.1
  local-id user1.roadw.6wind.net
  vip-pool user-vips
  security-policy hub
    local-ts subnet 192.168.0.0/24
    ..
  ..
pre-shared-key user1
  id user1@6wind.net
  secret 0seaJ31RfzHNRvUSH0oUYg7znTW0I=
  ..
pre-shared-key user2
  id user2@6wind.net
  secret 0s3zpRt+h3g12NSaSKEx2yjY4ctak=
  ..
pool user-vips
  address 192.168.99.0/24
  subnet 172.16.0.0/12
  subnet fc00:1234::/64
  ..
```

### Use case: route-based VPN concentrator

In this use case, remote users must be given access to the local site A via a public network. The traffic must be secured by IPsec VPNs between users and the security gateways SecGW-A. Dynamic SVTI interfaces and cross-VRF are used.



IKE negotiations are initiated by the remote users. Their public IP addresses are dynamically assigned by their access point. Each user requests the security gateway to assign it a virtual private address. The security gateway picks this virtual IP from a local pool.

The peers identify themselves with a user Fully Qualified Domain Name (user FQDN) and authenticate via pre-shared keys.

Plaintext traffic is in VRF private, while encrypted traffic is in vrf wan.

```
vrouter running config# show config nodelist / vrf private interface
interface
  physical eth1
    port pci-b0s4
    ipv4
      address 192.168.0.1/24
      ..
  ..
  svti-template svtemp
    mtu 1300
  ..
  ..

vrouter running config# show config nodelist / vrf wan interface
interface
  physical eth2
```

(continues on next page)

(continued from previous page)

```
port pci-b0s5
  ipv4
    address 192.0.2.1/24
    ..
  ..

vrouter running config# show config nodelist / vrf wan ike
ike
  global-options
    ..
  ike-policy-template iketemp1
    ike-proposal 1
      enc-alg aes128-cbc
      auth-alg hmac-sha512
      prf-alg hmac-sha512
      dh-group modp2048
      ..
    ..
  ipsec-policy-template ipsectemp1
    esp-proposal 1
      enc-alg aes128-cbc
      auth-alg hmac-sha256
      ..
    start-action none
    close-action none
    ..
  vpn siteA-roadw
    dynamic-svti
      svti-template svtitemp
      vrf private
    ike-policy
      template iketemp1
      ..
    ipsec-policy
      template ipsectemp1
      ..
    local-address 192.0.2.1
    local-id concentrator.6wind.net
    vip-pool user-vips
    security-policy hub
      local-ts subnet 192.168.0.0/24
      ..
    ..
  pre-shared-key user1
```

(continues on next page)

(continued from previous page)

```

    id user1@6wind.net
    secret 0seaJ31RfzHNRvUSH0oUYg7znTW0I=
    ..
pre-shared-key user2
    id user2@6wind.net
    secret 0s3zpRt+h3g12NSaSKEx2yjY4ctak=
    ..
pool user-vips
    address 192.168.99.0/24
    subnet 172.16.0.0/12
    subnet fc00:1234::/64
    ..

```

After a few negotiations and tear downs:

```

dut-vm running config# show state vrf private interface svti
svti dsvtiABJr_bNbVbc
    mtu 1300
    promiscuous false
    description "vpn:siteA-roadw remote-id:user3@6wind.net svti-template:svtitemp"
    enabled true
    svti-id 1
    link-vrf wan
    oper-status UNKNOWN
    counters
        in-octets 0
        in-unicast-pkts 0
        in-discards 0
        in-errors 0
        out-octets 0
        out-unicast-pkts 0
        out-discards 0
        out-errors 0
        ..
    link-interface lo
    ..
svti dsvtiABJsgSSWSNQ
    mtu 1300
    promiscuous false
    description "vpn:siteA-roadw remote-id:user42@6wind.net svti-template:svtitemp"
    enabled true
    svti-id 3
    link-vrf wan
    oper-status UNKNOWN

```

(continues on next page)

(continued from previous page)

```

counters
  in-octets 0
  in-unicast-pkts 0
  in-discards 0
  in-errors 0
  out-octets 0
  out-unicast-pkts 0
  out-discards 0
  out-errors 0
  ..
link-interface lo
..

```

```
dut-vm running config# show state vrf private routing static
```

```

static
  ipv4-route 192.168.0.3/32
    next-hop dsvtiABJr_bNbVbc
    ..
  ipv4-route 192.168.0.13/32
    next-hop dsvtiABJsgSSWSNQ
    ..
  (...)

```

```
dut-vm running config# show ike ike-sa vrf wan
```

| VPN              | Local Address | Local ID               | Remote Address | Remote ID        | State | IKE Version |
|------------------|---------------|------------------------|----------------|------------------|-------|-------------|
| ↳Child SA Count  |               |                        |                |                  |       |             |
| siteA-roadw      | 192.0.2.1     | concentrator.6wind.net | 10.125.0.5     | user3@6wind.net  | ↳     |             |
| ↳established 2 1 |               |                        |                |                  |       |             |
| siteA-roadw      | 192.0.2.1     | concentrator.6wind.net | 10.175.0.7     | user42@6wind.net | ↳     |             |
| ↳established 2 1 |               |                        |                |                  |       |             |

## Advanced configuration, performance and scalability

The base of the IKE control plane is the open source StrongSwan distribution.

In this section we focus on parameters useful to tune the scalability and performance of IKE.

## Logging

The IKE service is liable to issue many log messages. The verbosity of these logs is configurable per subsystem.

Messages issued by the IKE service are classified in 5 levels:

|   |  |
|---|--|
| 0 | Very basic auditing logs, (e.g. SA up/SA down)                         |
| 1 | Generic control flow with errors, a good default to see whats going on |
| 2 | More detailed debugging control flow                                   |
| 3 | Including RAW data dumps in hex  |
| 4 | Also include sensitive material in dumps, e.g. keys                    |

Messages may be issued by the following subsystems:

|          |  |
|----------|--|
| asn1     | Low-level encoding/decoding (ASN.1, X.509 etc.)            |
| child    | CHILD_SA/IPsec SA processing                               |
| config   | Configuration management and plugins                       |
| daemon   | Main daemon setup/cleanup/signal handling                  |
| encoding | Packet encoding/decoding encryption/decryption operations  |
| ike      | IKE_SA/ISAKMP SA processing                                |
| ipsec    | Libipsec library messages                                  |
| job      | Jobs queuing/processing and thread pool management         |
| kernel   | IPsec/Networking kernel interface                          |
| manager  | IKE_SA manager, handling synchronization for IKE_SA access |
| network  | IKE network communication                                  |

The logs may be sent to syslog facilities daemon and authpriv.

The default configuration for ike logs is the following:

```
vrouter running ike# show config logging
logging
  daemon
    default 0
    ..
  authpriv
    default disable
    ..
  ..
```

This configuration means that:

- messages of level 0 from all subsystems are sent to syslog facility daemon,
- no message from any subsystem is sent to syslog facility authpriv.

To alter this configuration, use the following command:

```
vrouter running ike# logging FACILITY SUBSYSTEM LEVEL
```

Where:

- FACILITY is the syslog facility (daemon or authpriv),
- SUBSYSTEM is the subsystem (see *IKE log subsystems*), or default to specify the default log level for all subsystems,
- LEVEL is the maximum log level of messages in the specified subsystem, (see *IKE log levels*) or disable to disable all messages,

### Example

The following commands modify which log messages are sent to facility authpriv:

- messages up to level 2 from the ike subsystem are logged to facility authpriv,
- messages up to level 1 from other subsystems are logged to facility authpriv.

```
vrouter running ike# logging
vrouter running logging# authpriv
vrouter running authpriv# default 1
vrouter running authpriv# ike 2
vrouter running authpriv# ..
vrouter running logging# ..
vrouter running ike#
vrouter running ike# show config logging
logging
  daemon
    default 0
    ..
  authpriv
    default 1
    ike 2
    ..
  ..
```

---

**Note:** Depending on the configuration, messages may be logged twice, once in facility daemon, and a second time in facility authpriv.

---

According to the configuration, log messages are sent to the daemon and/or authpriv syslog facilities with the notice severity. The severity is not configurable.

## Extended Sequence Number (ESN)

With throughputs getting higher and higher, the 32 bit IPsec sequence number may reach its maximum value before it is expected, so much that an Extended Sequence Number (ESN) option was defined (see **RFC 4304** (<https://tools.ietf.org/html/rfc4304.html>)), that extends the sequence number to 64 bits.

The use of ESN can be configured in each esp-proposal or ah-proposal in the ipsec-policy-template or vpn ipsec-policy. By default, ESN is disabled.

Require the use of ESN:

```
vrouter running ike# ipsec-policy-template ipsectemp1
vrouter running ipsec-policy-template ipsectemp1# esp-proposal 1
vrouter running esp-proposal 1# esn true
vrouter running esp-proposal 1# ..
vrouter running ipsec-policy-template ipsectemp1# ..
vrouter running ike#
```

```
vrouter running ike# show config nodefault
ike
  (...)
  ipsec-policy-template ipsectemp1
    (...)
    esp-proposal 1
      enc-alg aes128-cbc
      auth-alg hmac-sha256
      dh-group modp2048
      esn true
    ..
  ..
..
```

```
vrouter running ike# show config
ike
  (...)
  ipsec-policy-template ipsectemp1
    esp-proposal 1
      aead-alg aes128-gcm-128
      esn true
    ..
  ..
..
```

Refuse the use of ESN (default behavior):

```

vrouters running ike# ipsec-policy-template ipsectemp1
vrouters running ipsec-policy-template ipsectemp1# esp-proposal 1
vrouters running esp-proposal 1# esn false
vrouters running esp-proposal 1# ..
vrouters running ipsec-policy-template ipsectemp1# ..
vrouters running ike#

```

```

vrouters running ike# show config nodelist
ike
  (...)
  ipsec-policy-template ipsectemp1
    esp-proposal 1
      enc-alg aes128-cbc
      auth-alg hmac-sha256
      esn false
    ..
  ..
..

```

To specify that ESN is not mandatory but should be negotiated, specify both `esn true` and `esn false`, by order of preference:

```

vrouters running ike# ipsec-policy-template ipsectemp1
vrouters running ipsec-policy-template ipsectemp1# esp-proposal 1
vrouters running esp-proposal 1# esn true
vrouters running esp-proposal 1# esn false
vrouters running esp-proposal 1# ..
vrouters running ipsec-policy-template ipsectemp1# ..

```

```

vrouters running ike# show config
ike
  (...)
  ipsec-policy-template ipsectemp1
    esp-proposal 1
      enc-alg aes128-cbc
      auth-alg hmac-sha256
      esn true
      esn false
    ..
  ..
..

```

If no `esn` statement is specified, then ESN is disabled.

## Replay window size

There is no guarantee that IPsec packets are received by the security gateway in the same order as they were sent. With throughputs getting higher and higher, out-of-order IPsec packets may be dropped by the IPsec replay protection system if their lateness exceeds the replay window size. The size of the replay window can be increased to avoid such problem.

The replay window size option can be configured in the ipsec-policy-template (or vpn ipsec-policy):

```
vrouter running ike# ipsec-policy-template ipsectemp1
vrouter running ipsec-policy-template ipsectemp1# replay-window 4096
vrouter running ipsec-policy-template ipsectemp1# ..
vrouter running ike#
```

```
vrouter running ike# show config nodefault
ike
  (...)
  ipsec-policy-template ipsectemp1
    esp-proposal 1
      enc-alg aes128-cbc
      auth-alg hmac-sha256
      ..
    replay-window 4096
    ..
  ..
```

replay-window is an integer number of packets, in the range 0 to 4096 packets (default 32, 0 disables replay protection).

Note that the replay window size is a local choice, it does not impact the replay window size chosen by the remote peer.

## Virtual IP pools

IKEv1 and IKEv2 enable to assign a *virtual IP* during an IKE negotiation, i.e. an IKE initiator may request an additional IP address from the responder to use as inner IPsec tunnel address.

Virtual IPs are exchanged using the *mode config* extension in IKEv1, or using *configuration payloads* in IKEv2.

Additional parameters may be assigned during this exchange, such as a DNS server address, a NetBIOS server address or a DHCP server address.

To proceed, the responder maintains one or more pools of virtual IPs:

```
vrouter running vrf main# ike
vrouter running ike# pool my-pool
```

(continues on next page)

(continued from previous page)

```

vrouters running pool my-pool#! address 192.168.1.1-192.168.2.127
vrouters running pool my-pool# dns 192.168.3.99
vrouters running pool my-pool# nbns 192.168.3.99
vrouters running pool my-pool# dhcp 192.168.3.100
vrouters running pool my-pool# subnet 172.16.0.0/12
vrouters running pool my-pool# subnet fc00:1234::/64
vrouters running pool my-pool# ..
vrouters running ike#

```

- `address` is a list of addresses that can be assigned. Each list item can be a single address, a range of addresses or a subnet (IPv4 or IPv6).
- `dns` is an optional list of DNS server addresses (IPv4 or IPv6).
- `nbns` is an optional list of NetBIOS server addresses (IPv4 or IPv6).
- `dhcp` is an optional list of DHCP server addresses (IPv4 or IPv6).

A VPN can then reference a list of pools in its configuration:

```

vrouters running ike# vpn vpn-secgw
vrouters running vpn vpn-secgw# vip-pool my-pool
vrouters running vpn vpn-secgw# ..
vrouters running ike#

```

To include this dynamically assigned address in a security policy, make sure that no `remote-ts` is configured, or at least that the `remote-ts` subnet is unset (other fields such as the `protocol` may still be specified):

```

vrouters running ike# vpn vpn-secgw
vrouters running vpn vpn-secgw# security-policy dynamic-vip
vrouters running security-policy dynamic-vip# local-ts subnet 10.100.0.64/26
vrouters running security-policy dynamic-vip# remote-ts protocol 6
vrouters running security-policy dynamic-vip# ..
vrouters running vpn vpn-secgw# ..
vrouters running ike#

```

If an IKE initiator requests a virtual IP, it will be assigned one of the addresses in the `vip-pool(s)`, and the optional attributes (`dns`, `nbns`, `dhcp`).

## Retransmission constants

The IKE daemon uses an exponential backoff algorithm to calculate the timeout of packets before retransmission: the timeout grows exponentially with the number of tries, following the formula:

$$\text{timeout}_{\text{try}} = \text{retransmit-timeout} \times \text{retransmit-base}^{\text{try}}$$

Where try ranges from 0 to retransmit-tries. After retransmit-tries unsuccessful retransmissions, the IKE daemon gives up the negotiation.

The retransmission constants can be configured in the global-options section:

```
vrouter running ike# global-options
vrouter running global-options# retransmit-tries 3
vrouter running global-options# retransmit-timeout 3.0
vrouter running global-options# retransmit-base 1.0
vrouter running global-options# ..
vrouter running ike#
```

```
vrouter running ike# show config nodelist
ike
  (...)
  global-options
    (...)
    retransmit-tries 3
    retransmit-timeout 3.0
    retransmit-base 1.0
    ..
  ..
```

- retransmit-tries is an integer value ranging from 0 to 100 (default 5).
- retransmit-timeout is a decimal value ranging from 0.000 to 60.000 (default 4.0).
- retransmit-base is a decimal value ranging from 0.000 to 10.000 (default. 1.8).

For more information, see [strongSwan's IKE retransmission behavior](https://wiki.strongswan.org/projects/strongswan/wiki/Retr) (<https://wiki.strongswan.org/projects/strongswan/wiki/Retr>).

## Lifetime of SA acquire messages

By default IKE negotiations are triggered by outgoing traffic (ipsec-policy-template start-action trap).

When an outgoing packet matches a security policy that requires IPsec protection, but no suitable SA is available, an SA acquire message is raised to trigger the negotiation and a temporary IPsec SA is created in the IPsec stack.

This acquire SA prevents further acquire messages to be raised until the negotiation succeeds, or the acquire SA times out.

The default lifetime of an acquire SA is 165 seconds, this matches the total retransmission time of an IKE message that would receive no answer, with default retransmission constants.

This lifetime may be adjusted in the global-options section:

```
vrouter running ike# global-options
vrouter running global-options# acquire-timeout 60
vrouter running global-options# ..
vrouter running ike#
```

```
vrouter running ike# show config nodelist
ike
  (...)
  global-options
    (...)
    acquire-timeout 60
  ..
..
```

acquire-timeout is an integer number of seconds (default 165).

## DoS protection

The IKE daemon provides Deny of Service (DoS) protection using cookies and aggressiveness checks.

All DoS protection mechanisms are configured in the global-options dos-protection section.

```
vrouter running ike# global-options
vrouter running global-options# dos-protection
vrouter running dos-protection# cookie-threshold 12
vrouter running dos-protection# block-threshold 6
vrouter running dos-protection# init-limit-half-open 100
vrouter running dos-protection# ..
vrouter running global-options# ..
vrouter running ike#
```

```
vrouter running ike# show config nodelist
ike
  (...)
  global-options
    (...)
    dos-protection
      cookie-threshold 12
      block-threshold 6
      init-limit-half-open 100
```

(continues on next page)

(continued from previous page)

```

..
..
..

```

- **cookie-threshold** is the number of half-open IKE SAs that activate the cookie mechanism. It is an integer number or the keyword **always** (default 10). 0 disables the cookie mechanism. **always** activates it whatever the number of half-open SAs.
- **block-threshold** is the maximum number of half-open IKE SAs for a single peer IP. It is an integer number (default 5). 0 disables the limit.
- **init-limit-half-open** fixes a limit to the number of half open IKE SAs. New connections are refused if this limit is reached. It is an integer number (default 0). 0 disables the limit.

For more details, please refer to the `charon.cookie_threshold` and `charon.block_threshold` and `charon.init_limit_half_open` options in strongSwan's `strongswan.conf` configuration file (<https://wiki.strongswan.org/projects/strongswan/wiki/StrongswanConf#Defined-keys>).

## IKE worker threads

The IKE daemon is a multi-threaded application.

The total number of threads it uses may be configured in the `global-options` section.

```

vrouter running ike# global-options
vrouter running global-options# show config
vrouter running global-options# threads 20
vrouter running global-options# ..
vrouter running ike#

```

```

vrouter running ike# show config nodefault
ike
  (...)
  global-options
    (...)
    threads 20
  ..
..

```

`threads` is an 32 bit integer (default 16).

For more details, please refer to the `charon.threads` option in strongSwan's `strongswan.conf` configuration file (<https://wiki.strongswan.org/projects/strongswan/wiki/StrongswanConf#Defined-keys>).

## IKE SA hash table parameters

The IKE SA hash table size can be increased to improve performance when a high number of SAs is managed by the IKE daemon. It can be split into segments to improve performance when a high number of SAs is managed by the IKE daemon on multiple cores. Each segment will get its own lock.

It can be configured in the global-options section.

```
vrouter running ike# global-options
vrouter running global-options# sa-table-size 128
vrouter running global-options# sa-table-segments 16
vrouter running global-options# ..
vrouter running ike#
```

```
vrouter running ike# show config nodefault
ike
  (...)
  global-options
    (...)
    sa-table-size 128
    sa-table-segments 16
    ..
  ..
```

- `sa-table-size` is the size of the SA hash table (default 1).
- `sa-table-segments` is the number of segments (default 1).

For more details, please refer to the `charon.ikesa_table_size` option in `strongSwan's strongswan.conf` configuration file (<https://wiki.strongswan.org/projects/strongswan/wiki/StrongswanConf#Defined-keys>) and `strongSwan's` IKE SA lookup tuning (<https://wiki.strongswan.org/projects/strongswan/wiki/IkeSaTable>).

## IPsec SP hash table parameters

The IPsec security policy database (SPD) is an ordered list of rules, the security policies (SPs), that specify what IPsec processing must be applied to packets. They are composed of a packet selector (direction, source subnet, destination subnet, protocol, port) and an action (esp, ah, pass or drop). By default, these SPs are stored in a linked list. The time to browse this list increases with the number of SPs in  $O(n)$ .

When the IKE daemon establishes a child SA, it configures SPs in the IPsec stack. If the number of SPs grows, the time to add SPs grows in  $O(n)$ , which slows down the negotiation rate.

When the network stack processes traffic, it looks up for the IPsec policy to apply to outbound and inbound packets. If the number of SPs grows, the time to lookup for the right policy grows in  $O(n)$ , which slows down the throughput, regardless if packets need IPsec processing or not.

To solve this scalability issue, the IPsec stack maintains a hash table of security policies. SPs are hashed based on the source and destination address of their selector. These addresses are subnets with variable prefix lengths, which

prevents from hashing on all bits of the addresses. Some SPs cannot be hashed because their selector is too wide (the address prefix lengths are too small). These un-hashed SPs are stored in the linked list.

Thresholds are defined, to select which SPs will be hashed and how many bits of address will be included in the hash key:

```
vrouter running ike# global-options
vrouter running global-options# sp-hash-ipv4 local 16 remote 24
vrouter running global-options# sp-hash-ipv6 local 56 remote 64
vrouter running global-options# ..
vrouter running ike#
```

```
vrouter running ike# show config nodefault
ike
  (...)
  global-options
    (...)
    sp-hash-ipv4 local 16 remote 24
    sp-hash-ipv6 local 56 remote 64
  ..
```

- `sp-hash-ipv4 local` and `remote` are the local and remote address minimum prefix lengths of hashed IPv4 SPs. They range from 0 to 32 (default 32).
- `sp-hash-ipv6 local` and `remote` are the local and remote address minimum prefix lengths of hashed IPv6 SPs. They range from 0 to 128 (default 128).

SPs whose local and remote address prefix lengths are greater or equal to the thresholds are hashed (which speeds up the lookup and insertion), others are simply looked up in sequence. For hashed SPs, the high order bits of the address (up to the threshold) are included in the hash key calculation.

Example:

```
dir out src 10.22.0.0/20 dst 10.24.1.0/24 => hashed
dir out src 10.22.0.0/16 dst 10.24.0.0/16 => unhashed
dir in  src 10.24.1.1/32 dst 10.22.0.0/16 => hashed

dir out src 3ffe:304:124:2200::/60 dst 3ffe:304:124:2401::/64 => hashed
dir out src 3ffe:304:124:2200::/56 dst 3ffe:304:124:2400::/56 => unhashed
dir in  src 3ffe:304:124:2401::2/128 dst 3ffe:304:124:2200::/56 => hashed
```

Hash thresholds not only determine which policies will be hashed, but also the number of bits of the local and remote address that will be used to calculate the hash key. Big thresholds mean potentially fewer hashed policies, but better distribution in the hash table, and vice versa.

A good trade off must be found depending on the prefix lengths used in the SPD.

## Reverse route injection

Routes can be inserted into a separate routing table for established IPsec tunnels. This enables to inject routes to the remote network discovered during an IKE negotiation.

```
vrouter running ike# global-options
vrouter running global-options# install-routes true
vrouter running global-options# routing-table 230
vrouter running global-options# routing-table-prio 230
vrouter running global-options# ..
vrouter running ike#
```

```
vrouter running ike# show config nodefult
ike
  (...)
  global-options
    (...)
    install-routes true
    routing-table 230
    routing-table-prio 230
  ..
```

- `install-routes` activates or deactivates route installation (default false).
- `routing-table` is the number of the routing table in which routes will be injected (Default 220).
- `routing-table-prio` is the priority of the Policy-Based Routing (PBR) rule that requests to lookup in the routing table (default 220).

## IKEv2 Mobility and Multihoming Protocol (MOBIKE)

MOBIKE (RFC 4555) allows the IP addresses associated with IKEv2 and tunnel mode IPsec Security Associations to change. A mobile Virtual Private Network (VPN) client could use MOBIKE to keep the connection with the VPN gateway active while moving from one address to another. Similarly, a multihomed host could use MOBIKE to move the traffic to a different interface if, for instance, the one currently being used stops working.

MOBIKE can be enabled in the IKE policy template:

```
vrouter running config# / vrf main ike
vrouter running ike# ike-policy-template my_policy_tmpl
vrouter running ike-policy-template my_policy_tmpl# mobike true
```

Alternatively, it can be enabled in the vpn ike policy:

```
vrouter running config# / vrf main ike vpn my_vpn
vrouter running vpn my_vpn#! ike-policy template my_policy_tmpl
```

(continues on next page)

(continued from previous page)

```
vrouter running vpn my_vpn#! ipsec-policy template my_ipsec_tmpl
vrouter running vpn my_vpn# ike-policy mobike true
```

By default, when MOBIKE is enabled, the SA addresses are not modified if the routing path is still usable. Enabling `mobike-prefer-best-path` in global options dynamically changes this behavior: on routing change, if a cheaper path exists, the SA will be updated dynamically.

To enable the `mobike-prefer-best-path` option:

```
vrouter running ike# global-options
vrouter running global-options# mobike-prefer-best-path true
```

**See also:**

The *command reference*.

## IP packet filtering

This is where IPv4 and IPv6 packet filtering is configured. The device monitors incoming and outgoing traffic, and determines whether to allow or deny traffic, based on sequenced list of rules. Each rule contains a packet selector and the related action.

The IP packet filtering module leverages Netfilter, and re-uses its concepts.

---

**Note:** Filtering rules not configured by the management system will not be displayed by `show state` and will be lost when a new firewall configuration is committed.

---

**Caution:** Stateful IP packet filtering and CG-NAT are exclusive. If CG-NAT is enabled, stateful IP packet filtering must be disabled on ports bound to the fast path.

**See also:**

The *command reference* for details.

- *Definitions*
  - *Chains*
  - *Tables*
  - *Rules*
  - *Groups*
  - *Connection tracking*

- *Stateless filtering*
- *Stateful filtering*

## Definitions

### Chains

A chain is a list of rules. It is responsible for determining how an incoming, outgoing or forwarded packet should be processed by the filtering module.

There are several default chains, associated to hooks in the routing stack:

- `prerouting` is called as soon as packets are received
- `input` is called for locally delivered packets
- `forward` is called when the packet is being routed
- `output` is called for locally generated packets
- `postrouting` is called when the packets are about to be sent

If a packet entering a default chain does not match any rule, it will be processed by the chain's default policy.

User-chains can be defined as well, and called from the default chains.

### Tables

Several tables are available. Each table has a specific purpose and defines some specific default chains. The chains cross the different tables in a predefined priority.

The `raw` table is mainly used to exempt packets from connection tracking. Only the `prerouting` and `output` chains are available in this table. It is always crossed first (before connection tracking).

The `mangle` table is the packet alteration table. All the default chains are available in this table. It is called before filter, and after connection tracking.

The `filter` table is the default table. Only the `input`, `forward` and `output` chains are available in this table. This is where packets are actually filtered. It is called after the `mangle` table.

## Rules

A rule is defined by a sequence number, a packet selector and an action. It specifies what action to apply to packets that match the selector. Packets are compared to each rule until it matches one rule selector. The action is then applied to the packet and look up into the chain is stopped. If a packet does not match any of the rules in a default chain, it is applied the default policy.

## Groups

A group is a set of IP addresses or networks. The rule packet selector can reference a group as source or destination.

## Connection tracking

To perform stateful filtering or NAT, the device can monitor the connections and maintain their state, using the connection tracking module. Each connection is stored in a `conntrack`, defined by its source and destination address, its source and destination port in the two directions (named origin and reply), and the state of the connection.

Here is an example of a `conntrack` defining an SSH connection from 10.0.2.2 port 58242 to 10.0.2.15 port 22. The connection is established, meaning that packets were seen in the two directions.

```
tcp      6 431995 ESTABLISHED src=10.0.2.2 dst=10.0.2.15 sport=58242 dport=22 src=10.0.
↔2.15 dst=10.0.2.2 sport=22 dport=58242 [ASSURED] mark=0 use=1
```

The connection tracking module is called in prerouting and output chains, after the `raw` table. It is enabled for all packets, unless disabled using the `notrack` action.

The `show conntracks` command displays the currently created `conntracks`.

```
vrouter> show conntracks
tcp      6 431995 ESTABLISHED src=10.0.2.2 dst=10.0.2.15 sport=58242 dport=22 src=10.0.
↔2.15 dst=10.0.2.2 sport=22 dport=58242 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
```

## Stateless filtering

Stateless filtering does not need the connection tracking module.

Let's configure the following:

- create a user chain named `outside` to store rules common to the public interfaces `pub0` and `pub1`
- change the input policy to `drop`
- create a `trusted` address group containing the `2.2.2.2` and `4.4.4.4` IP addresses

- allow all traffic from 1.1.1.1 IP address and trusted group, and only ssh and netconf connections from other IPs entering from the pub0 and pub1 interfaces
- allow all the traffic entering from the priv interface
- allow all the traffic entering from the lo interface (used by the cli)

```

vrouter running vrf main# group ipv4
vrouter running ipv4# address-group trusted
vrouter running address-group trusted# address 2.2.2.2
vrouter running address-group trusted# address 4.4.4.4
vrouter running address-group trusted# .. .. ..
vrouter running vrf main# firewall ipv4
vrouter running ipv4# filter
vrouter running filter# chain outside
vrouter running chain outside# rule 1 source address 1.1.1.1 description "allow 1.1.1.1
↳" action accept
vrouter running chain outside# rule 2 source group trusted description "allow trusted"
↳action accept
vrouter running chain outside# rule 3 protocol tcp destination port 22 description
↳"allow ssh" action accept
vrouter running chain outside# rule 4 protocol tcp destination port 830 description
↳"allow netconf" action accept
vrouter running chain outside# ..
vrouter running filter# input
vrouter running input# policy drop
vrouter running input# rule 1 inbound-interface pub0 action chain outside
vrouter running input# rule 2 inbound-interface pub1 action chain outside
vrouter running input# rule 3 inbound-interface priv action accept
vrouter running input# rule 4 inbound-interface lo description "allow local netconf
↳traffic" action accept

```

---

**Note:** This configuration is partial, and only shown as an example. It should not be used as is in production.

---

Let's fetch the state after committing this configuration:

```

vrouter running vrf main# show state group
group
  ipv4
    address-group trusted
      address 2.2.2.2
      address 4.4.4.4
      ..
    ..
  ..

```

(continues on next page)

(continued from previous page)

```

vrouters running vrf main# show state firewall ipv4 filter
filter
  input
    bytes 23862
    policy drop
    packets 111
    rule 1 counters bytes 0 packets 0 inbound-interface pub0 action chain outside
    rule 2 counters bytes 0 packets 0 inbound-interface pub1 action chain outside
    rule 3 counters bytes 0 packets 0 inbound-interface priv action accept
    rule 4 description "allow local netconf traffic" counters bytes 803700 packets 2289
    inbound-interface lo action accept
    ..
  forward
    bytes 0
    policy accept
    packets 0
    ..
  output
    bytes 811590
    policy accept
    packets 2400
    ..
  chain outside
    bytes 0
    packets 0
    rule 1 description "allow 1.1.1.1" counters bytes 0 packets 0 source address 1.1.1.1/32
    action accept
    rule 2 description "allow trusted" counters bytes 0 packets 0 source group trusted
    action accept
    rule 3 description "allow ssh" counters bytes 0 packets 0 protocol tcp destination
    port 22 action accept
    rule 4 description "allow netconf" counters bytes 0 packets 0 protocol tcp destination
    port 830 action accept
    ..
  ..

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouters> show config xml absolute vrf main group
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <group xmlns="urn:6wind:vrouter/group">
      <ipv4>

```

(continues on next page)

(continued from previous page)

```

    <address-group>
      <name>trusted</name>
      <address>2.2.2.2</address>
      <address>4.4.4.4</address>
    </address-group>
  </ipv4>
</group>
</vrf>
</config>

```

**vrouter>** show config xml absolute vrf main firewall

```

<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <firewall xmlns="urn:6wind:vrouter/firewall">
      <ipv4>
        <filter>
          <forward>
            <policy>accept</policy>
          </forward>
          <output>
            <policy>accept</policy>
          </output>
          <input>
            <policy>drop</policy>
          </input>
          <rule>
            <id>1</id>
            <inbound-interface>
              <name>pub0</name>
            </inbound-interface>
            <action>
              <chain>outside</chain>
            </action>
          </rule>
          <rule>
            <id>2</id>
            <inbound-interface>
              <name>pub1</name>
            </inbound-interface>
            <action>
              <chain>outside</chain>
            </action>
          </rule>
          <rule>

```

(continues on next page)

(continued from previous page)

```
<id>3</id>
<inbound-interface>
  <name>priv</name>
</inbound-interface>
<action>
  <standard>accept</standard>
</action>
</rule>
<rule>
  <id>4</id>
  <inbound-interface>
    <name>lo</name>
  </inbound-interface>
  <description>allow local netconf traffic</description>
  <action>
    <standard>accept</standard>
  </action>
</rule>
</input>
<chain>
  <name>outside</name>
  <policy>accept</policy>
  <rule>
    <id>1</id>
    <source>
      <address>
        <value>1.1.1.1</value>
      </address>
    </source>
    <description>allow 1.1.1.1</description>
    <action>
      <standard>accept</standard>
    </action>
  </rule>
  <rule>
    <id>2</id>
    <source>
      <group>
        <value>trusted</value>
      </group>
    </source>
    <description>allow trusted</description>
    <action>
      <standard>accept</standard>
```

(continues on next page)

(continued from previous page)

```
    </action>
  </rule>
  <rule>
    <id>3</id>
    <protocol>
      <value>tcp</value>
    </protocol>
    <destination>
      <port>
        <value>22</value>
      </port>
    </destination>
    <description>allow ssh</description>
    <action>
      <standard>accept</standard>
    </action>
  </rule>
  <rule>
    <id>4</id>
    <protocol>
      <value>tcp</value>
    </protocol>
    <destination>
      <port>
        <value>830</value>
      </port>
    </destination>
    <description>allow netconf</description>
    <action>
      <standard>accept</standard>
    </action>
  </rule>
</chain>
</filter>
</ipv4>
</firewall>
</vrf>
</config>
```

## Stateful filtering

Using the connection tracking, it is possible to match packets that are part of an existing connection.

The following configuration adds to the previous stateless configuration some stateful filtering rules, by allowing packets from an existing connection, but denying packets for a new one.

```
vrouter running vrf main# firewall ipv4
vrouter running ipv4# filter
vrouter running filter# chain outside
vrouter running chain outside# rule 5 conntrack state established related description
↳ "accept established and related connections" action accept
vrouter running chain outside# rule 6 conntrack state new description "deny new
↳ connections" action drop
vrouter running chain outside# commit
```

### 3.1.10 High Availability

#### High-availability Groups

A high-availability group is used to list a set of services whose state (*master* or *backup*) switch together.

The state of the high-availability group can be defined in the configuration, or it can be driven by another service (for instance, *vrp*) which declares itself as a controller for this high-availability group. There is one and only one controller for a group.

Some services like *ike* can subscribe to this high-availability group to be notified when the state of the group changes. A group can have several subscribers.

To create a high-availability group called *my-ha-group*, statically controlled by configuration:

```
vrouter running config# ha group my-ha-group
vrouter running group my-ha-group#! state master
vrouter running group my-ha-group# commit
```

The `state` command defines the administrative state of this group. If it is omitted, the state has to be driven by another service.

Let's fetch the state after committing this configuration:

```
vrouter running group my-ha-group# show state
group my-ha-group
  state master
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```

vrouter running config# show config xml absolute ha group
<config xmlns="urn:6wind:vrouter">
  <ha xmlns="urn:6wind:vrouter/ha">
    <group>
      <name>my-ha-group</name>
      <state>master</state>
    </group>
  </ha>
</config>

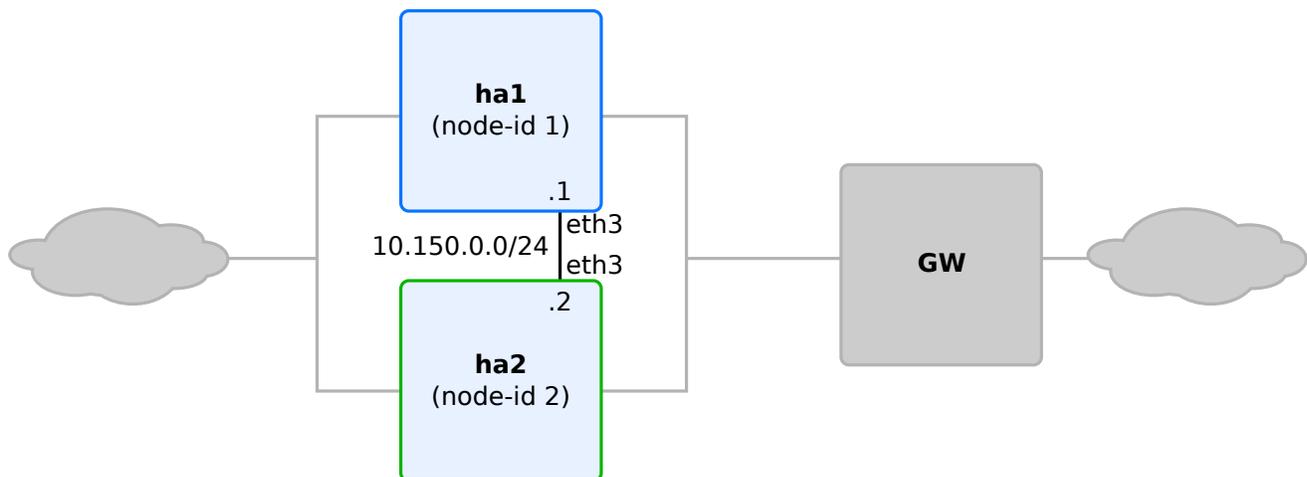
```

## High Availability neighbor

High Availability neighbor enables synchronizing ARP/NDP entries between two HA nodes in master/backup mode.

If the activity is switched between the two nodes, the new active node will be able to take over dataplane traffic and synchronize its new ARP/NDP entries with the inactive node.

The activity of a node can be controlled by CLI commands or by external applications (such as the VRRP service).



In this example, ARP/NDP entries learned by the *ha1* (the master) will be sent to *ha2* (the backup) through *eth3*.

HA neighbor parameters are configured in the *ha-neighbor* context:

```

ha1 running config# vrf main ha-neighbor
ha1 running ha-neighbor#!

```

Configure the *ha1*:

```
ha1 running ha-neighbor#! node-id 1
ha1 running ha-neighbor#! interface eth3
ha1 running ha-neighbor#! local-address 10.150.0.1
ha1 running ha-neighbor#! listen-ha-group ha-group1
ha1 running ha-neighbor#
```

- `node-id` is a unique identifier for this node in the HA cluster. It ranges from 1 to 15.
- `interface` is the network interface on which synchronization packets are exchanged
- `local-address` is the IPv4 or IPv6 addresses of the other HA node.
- `listen-ha-group` is the high-availability group that controls the activity state of this HA node. See *High-availability Groups* for more information.

Display *ha1* HA neighbor state:

```
ha1 running ha-neighbor# show state
ha-neighbor
  enabled true
  node-id 1
  local-address 10.150.0.1
  listen-ha-group ha-group1
  interface eth3
  state master
  ..
```

On *ha2*, the `node-id`, `interface` and the `local-address` must be adjusted:

```
ha2 running config# vrf main ha-neighbor
ha2 running ha-neighbor#! node-id 2
ha2 running ha-neighbor#! interface eth3
ha2 running ha-neighbor#! local-address 10.150.0.2
ha2 running ha-neighbor#! listen-ha-group ha-group1
ha2 running ha-neighbor#
```

Display *ha2* HA neighbor state:

```
ha2 running ha-neighbor# show state
ha-neighbor
  enabled true
  node-id 2
  local-address 10.150.0.2
  listen-ha-group ha-group1
  interface eth3
  state backup
  ..
```

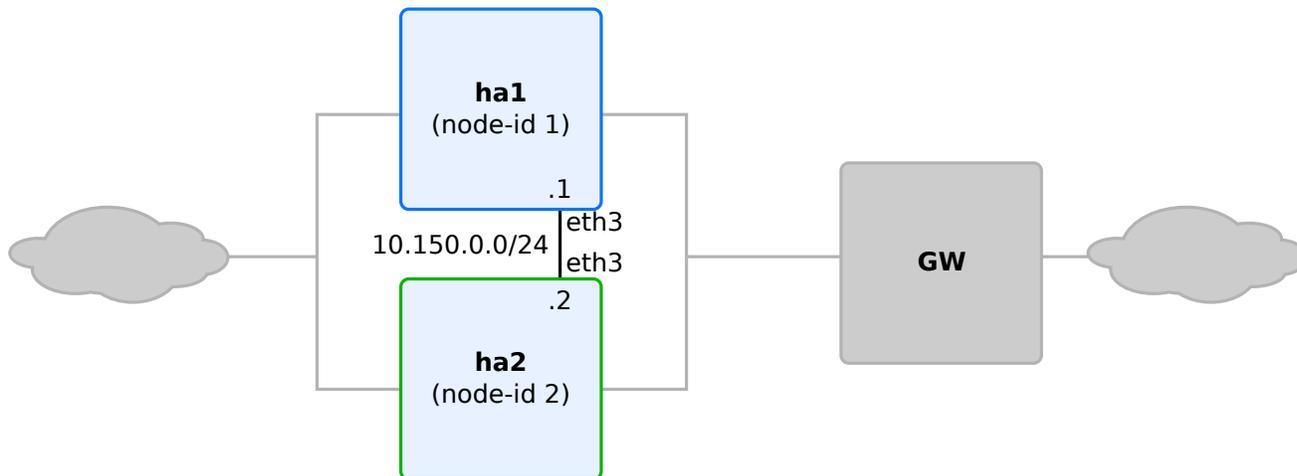
## High Availability contrack

High Availability contrack enables synchronizing contracks between two or more HA nodes in master/backup mode.

It maintains an internal cache (reflecting the contracks in the dataplane) and an external cache (contracks advertised by the other HA node). The content of the caches is used only when the system changes HA states:

- when switching to master, these 4 steps are executed:
  - commit the external cache into the dataplane
  - flush the internal and the external caches
  - resynchronize the internal cache to the dataplane
  - then send a bulk update to backups
- when switching to backup:
  - shorten dataplane contrack timers to remove the zombie entries
  - request resynchronization with master firewall replica (if any)

The activity of a node can be controlled by CLI commands or by external applications (such as the VRRP service).



In this example, contracks are synchronized from *ha1* to *ha2* external table as time goes along, and configured in *ha2*'s dataplane when *ha2* becomes master.

HA contrack parameters are configured per VRF in the `ha-contrack` context:

```
ha1 running config# vrf main ha-contrack
ha1 running ha-contrack#!
```

Configure mandatory options in *ha1*:

```
ha1 running ha-contrack#! interface eth3
ha1 running ha-contrack#! local-address 10.150.0.1
ha1 running ha-contrack#! listen-ha-group ha-group1
ha1 running ha-contrack#
```

- `interface` is the interface to use to send the synchronization messages.
- `local-address` is the IPv4 or IPv6 addresses of the interface used for the synchronization.
- `listen-ha-group` is the high-availability group that controls the activity state of this HA node. See *High-availability Groups* for more information.

---

**Note:** Protocols and IP addresses events can also be filtered respectively through `protocol-list` and `address-list` options. This filter can be an include or exclude logic depending on the `accept true|false` option value. See the *HA contrack command reference* for details.

---

Display *ha1* HA contrack state:

```
ha1 running ha-contrack# show state
ha-contrack
  enabled true
  local-address 10.150.0.1
  listen-ha-group ha-group1
  interface eth3
  state master
  ..
```

On *ha2*, the interface and the local-address must be adjusted:

```
ha2 running config# vrf main ha-neighbor
ha2 running ha-contrack#! interface eth3
ha2 running ha-contrack#! local-address 10.150.0.2
ha2 running ha-contrack#! listen-ha-group ha-group1
ha2 running ha-contrack#
```

Display *ha2* HA contrack state:

```
ha2 running ha-contrack# show state
ha-contrack
  enabled true
  local-address 10.150.0.2
  listen-ha-group ha-group1
  interface eth3
```

(continues on next page)

(continued from previous page)

```
state backup
..
```

---

**Note:** High Availability IKE requires a Turbo IPsec Application License.

---

## High Availability IKE

High Availability Internet Key Exchange (HA IKE) is an IKE extension that enables to perform stateful synchronization of IKE between two HA nodes in active/backup mode.

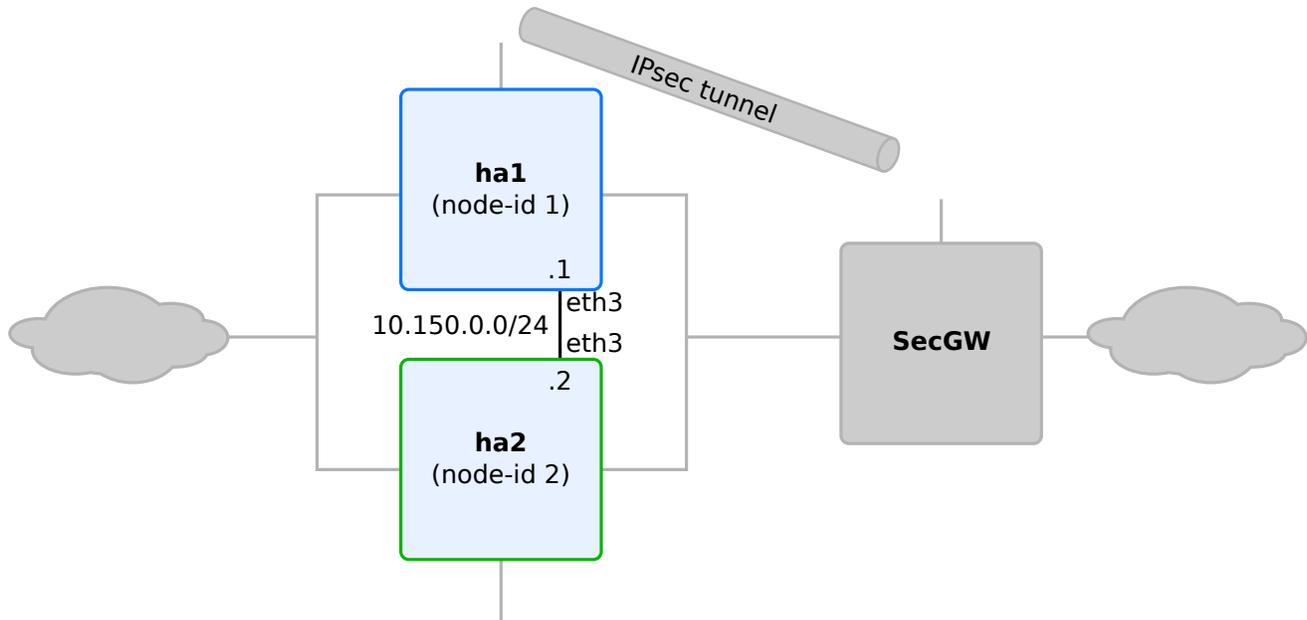
HA IKE may be configured between two nodes forming an HA cluster: the IKE internal states (IKE SAs and CHILD SAs) and IPsec SAs sequence numbers are synchronized from the active node to the backup node.

If the activity is switched between the two nodes, the new active node will be able to take over the IKE negotiations and IPsec dataplane traffic.

- *Overview*
- *Use case: HA IKE cluster with VRRP*
  - *ha1 CLI configuration*
  - *ha2 CLI configuration*
- *Advanced options*
  - *Sequence number synchronization parameters*
  - *HA-compatible virtual IP pools*

## Overview

The activity of a node can be controlled by CLI commands or by external applications (such as the VRRP service).



HA IKE parameters are configured in the **ha** sub-context of **ike**.

Enter the **ha** sub-context on **ha1**:

```
ha1 running config# vrf main
ha1 running vrf main# ike
ha1 running ike# ha
ha1 running ha#!
```

Configure HA IKE parameters:

```
ha1 running ha#! node-id 1
ha1 running ha#! interface eth3
ha1 running ha#! local-address 10.150.0.1
ha1 running ha#! remote-address 10.150.0.2
ha1 running ha#! listen-ha-group ha-group1
ha1 running ha#
```

- **node-id** is a unique identifier for this node in the HA cluster. It ranges from 1 to 15.
- **interface** is the network interface on which synchronization packets are exchanged
- **local-address** and **remote-address** are the IPv4 or IPv6 addresses of the two HA nodes.
- **listen-ha-group** is the high-availability group that controls the activity state of this HA node. See *High-availability Groups* for more information.

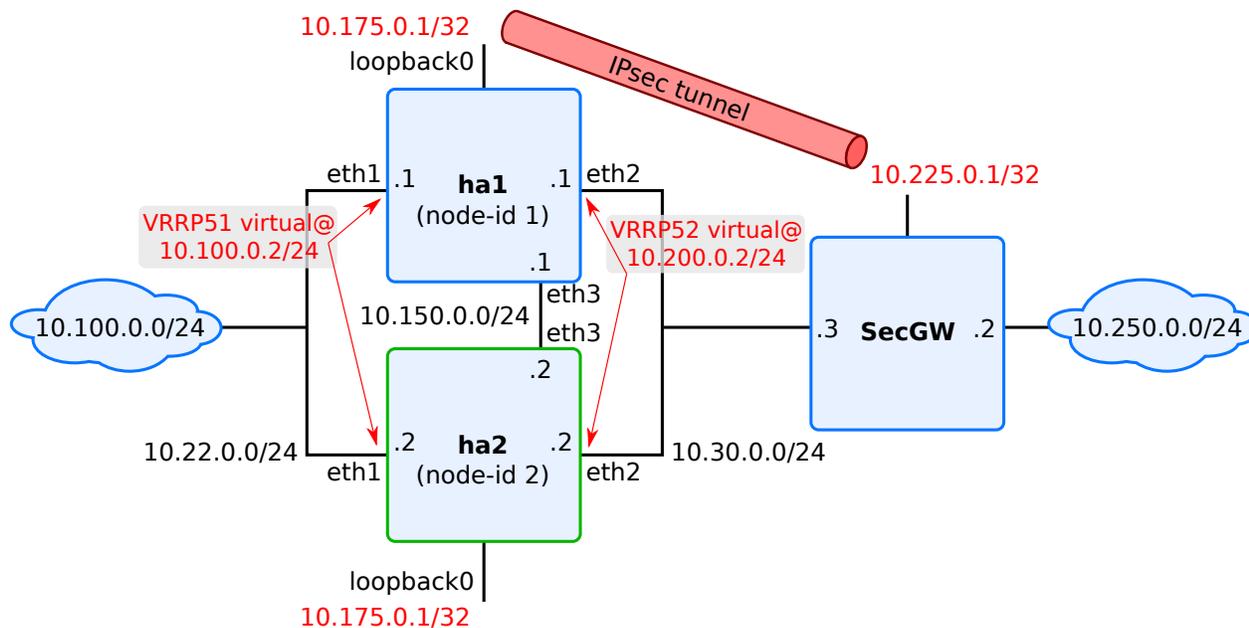
Display HA IKE parameters:

```
ha1 running ha# show config
ha
  enabled true
  listen-ha-group my-ha-group
  node-id 1
  interface eth3
  local-address 10.150.0.1
  remote-address 10.150.0.2
  seqnum-sync
    oseq-shift 65536
    sync-period-time 10s
    sync-period-packets 2
  ..
..
```

On *ha2*, the *node-id* and *interface* must be adjusted and the *local-address* and *remote-address* swapped:

```
ha2 running config# vrf main ike ha
ha2 running ha#! node-id 2
ha2 running ha#! interface eth3
ha2 running ha#! local-address 10.150.0.2
ha2 running ha#! remote-address 10.150.0.1
ha2 running ha#! listen-ha-group ha-group1
```

## Use case: HA IKE cluster with VRRP



In this use case, two devices *ha1* and *ha2* are configured as a redundant security gateway, performing IKE negotiations with a remote security gateway *SecGW*.

The activity of each HA node is determined by the VRRP protocol (see *VRRP command reference* for details about VRRP).

The two HA devices must be configured exactly the same, except for HA parameters (VRRP and HA IKE).

### ha1 CLI configuration

Configure device hostname:

```
vrouter running config# system hostname ha1
vrouter running config# commit
Configuration committed.
```

Configure the HA group:

```
vrouter running config# ha group ha-group1
ha1 running group ha-group1#! ..
ha1 running ha#! ..
```

**Note:** The ha-group maintains the node high-availability state. It is controlled by the VRRP protocol (via the

notify-ha-group command) and monitored by HA IKE (via the listen-ha-group command). Only one controller can be defined for an ha-group.

---

Move to vrf main configuration:

```
ha1 running config#! vrf main
ha1 running vrf main#!
```

Configure the network interfaces (adapt port ids to your hardware):

```
ha1 running vrf main#! interface physical eth1
ha1 running physical eth1#! port pci-b0s3
ha1 running physical eth1#! ipv4 address 10.22.0.1/24
ha1 running physical eth1#! ..
ha1 running interface#! physical eth2
ha1 running physical eth2#! port pci-b0s4
ha1 running physical eth2#! ipv4 address 10.23.0.1/24
ha1 running physical eth2#! ..
ha1 running interface#! physical eth3
ha1 running physical eth3#! port pci-b0s5
ha1 running physical eth3#! ipv4 address 10.150.0.1/24
ha1 running physical eth3#! ..
ha1 running interface#! loopback loopback0
ha1 running loopback loopback0#! ipv4 address 10.175.0.1/32
ha1 running loopback loopback0#! ..
ha1 running interface#! ..
```

Configure routes:

```
ha1 running vrf main#! routing
ha1 running routing#! static
ha1 running static#! ipv4-route 10.250.0.0/24 next-hop 10.200.0.1
ha1 running static#! ipv4-route 10.225.0.0/24 next-hop 10.200.0.1
ha1 running static#! ..
ha1 running routing#! ..
```

Configure VRRP:

```
ha1 running vrf main#! interface vrrp vrrp51
ha1 running vrrp vrrp52#! vrid 1
ha1 running vrrp vrrp51#! link-interface eth1
ha1 running vrrp vrrp51#! priority 100
ha1 running vrrp vrrp51#! advertisement-interval 1000
ha1 running vrrp vrrp51#! virtual-address 10.100.0.2/24
ha1 running vrrp vrrp51#! ..
```

(continues on next page)

(continued from previous page)

```

ha1 running interface#! vrrp vrrp52
ha1 running vrrp vrrp52#! vrid 1
ha1 running vrrp vrrp52#! link-interface eth2
ha1 running vrrp vrrp52#! priority 100
ha1 running vrrp vrrp52#! advertisement-interval 1000
ha1 running vrrp vrrp52#! virtual-address 10.200.0.2/24
ha1 running vrrp vrrp52#! ..
ha1 running interface#! ..
ha1 running vrf main#! vrrp group group1
ha1 running group group1#! instance vrrp51
ha1 running group group1#! instance vrrp52
ha1 running group group1#! notify-ha-group ha-group1
ha1 running group group1# ..
ha1 running vrrp# ..

```

Show the configuration:

```

ha1 running vrf main# show config nodefult
vrf main
  interface
    vrrp vrrp51
      link-interface eth1
      vrid 1
      virtual-address 10.100.0.2/24
      ..
    vrrp vrrp52
      link-interface eth2
      vrid 1
      virtual-address 10.200.0.2/24
      ..
  ..
vrrp
  group group1
    instance vrrp51
    instance vrrp52
    notify-ha-group ha-group1
    ..
  ..
..

```

Configure IKE:

```

ha1 running vrf main# ike
ha1 running ike# ike-policy-template ike1

```

(continues on next page)

(continued from previous page)

```

ha1 running ike-policy-template ike1# ike-proposal 1 enc-alg aes128-cbc auth-alg hmac-
↳sha1 dh-group modp1024
ha1 running ike-policy-template ike1# rekey-time 2h
ha1 running ike-policy-template ike1# ..
ha1 running ike# ipsec-policy-template ipsec1
ha1 running ipsec-policy-template ipsec1# esp-proposal 1 enc-alg aes128-cbc auth-alg_
↳hmac-sha1 esn true
ha1 running ipsec-policy-template ipsec1# rekey-time 1h
ha1 running ipsec-policy-template ipsec1# replay-window 1024
ha1 running ipsec-policy-template ipsec1# ..
ha1 running ike# vpn vpn-secgw
ha1 running vpn vpn-secgw#! ike-policy template ike1
ha1 running vpn vpn-secgw#! ipsec-policy template ipsec1
ha1 running vpn vpn-secgw# local-address 10.175.0.1
ha1 running vpn vpn-secgw# remote-address 10.225.0.1
ha1 running vpn vpn-secgw# security-policy site-to-secgw-site
ha1 running security-policy site-to-secgw-site# local-ts subnet 10.100.0.64/26
ha1 running security-policy site-to-secgw-site# remote-ts subnet 10.250.0.192/26
ha1 running security-policy site-to-secgw-site# ..
ha1 running vpn vpn-secgw# ..
ha1 running ike# pre-shared-key secgw
ha1 running pre-shared-key secgw#! id 10.225.0.1
ha1 running pre-shared-key secgw#! secret 0sBzAyaM5PTcnTHi/yRA1lARpAoRetSzP8
ha1 running pre-shared-key secgw# ..
ha1 running ike#

```

Show IKE configuration:

```

ha1 running ike# show config nodefault
ike
  pre-shared-key secgw
    id 10.225.0.1
    secret 0sBzAyaM5PTcnTHi/yRA1lARpAoRetSzP8
    ..
  global-options
    dos-protection
    ..
    sp-hash-ipv4
    sp-hash-ipv6
    ..
  ike-policy-template ike1
    ike-proposal 1
      enc-alg aes128-cbc
      auth-alg hmac-sha1

```

(continues on next page)

(continued from previous page)

```

        dh-group modp1024
        ..
    rekey-time 2h
    ..
ipsec-policy-template ipsec1
    esp-proposal 1
        enc-alg aes128-cbc
        auth-alg hmac-sha1
        esn true
        ..
    replay-window 1024
    ..
vpn vpn-secgw
    ike-policy
        template ike1
        ..
    ipsec-policy
        template ipsec1
        ..
    local-address 10.175.0.1
    remote-address 10.225.0.1
    security-policy site-to-secgw-site
        local-ts subnet 10.100.0.64/26
        remote-ts subnet 10.250.0.192/26
        ..
    ..
..

```

Configure HA IKE:

```

ha1 running ike# ha
ha1 running ha#! node-id 1
ha1 running ha#! interface eth3
ha1 running ha#! local-address 10.150.0.1
ha1 running ha#! remote-address 10.150.0.2
ha1 running ha#! listen-ha-group ha-group1
ha1 running ha# ..
ha1 running ike# commit
Configuration committed.
ha1 running ike#

```

Show HA IKE configuration:

```

ha1 running ike# show config nodefaut ha

```

(continues on next page)

(continued from previous page)

```
ha
  listen-ha-group ha-group1
  node-id 1
  interface eth3
  local-address 10.150.0.1
  remote-address 10.150.0.2
  seqnum-sync
  ..
..
```

## ha2 CLI configuration

A similar configuration is used for *ha2*. The differences are the hostname, the physical interfaces addresses, VRRP parameters and IKE HA parameters.

The IKE parameters (except HA ones) must be strictly identical.

```
ha2 running config# show config nodefault
config
  vrf main
    interface
      physical eth1
        ipv4
          address 10.22.0.2/24
          ..
        ..
      physical eth2
        ipv4
          address 10.23.0.2/24
          ..
        ..
      physical eth3
        ipv4
          address 10.150.0.2/24
          ..
        ..
      loopback loopback0
        ipv4
          address 10.175.0.1/32
          ..
        ..
      vrrp vrrp51
        link-interface eth1
```

(continues on next page)

(continued from previous page)

```
        vrid 1
        virtual-address 10.100.0.2/24
        ..
    vrrp vrrp52
        link-interface eth2
        vrid 1
        virtual-address 10.200.0.2/24
        ..
    ..
routing
    static
        ipv4-route 10.250.0.0/24
            next-hop 10.200.0.1
            ..
        ipv4-route 10.225.0.0/24
            next-hop 10.200.0.1
            ..
        ..
    ..
vrrp
    group group1
        instance vrrp51
        instance vrrp52
        notify-ha-group ha-group1
        ..
    ..
ike
    pre-shared-key secgw
        id 10.225.0.1
        secret 0sBzAyaM5PTcnTHi/yRA1lARpAoRetSzP8
        ..
    global-options
        dos-protection
        ..
        sp-hash-ipv4
        sp-hash-ipv6
        ..
    ha
        listen-ha-group ha-group1
        node-id 2
        interface eth3
        local-address 10.150.0.2
        remote-address 10.150.0.1
        seqnum-sync
```

(continues on next page)

(continued from previous page)

```
..
..
ike-policy-template ike1
  ike-proposal 1
    enc-alg aes128-cbc
    auth-alg hmac-sha1
    dh-group modp1024
    ..
  rekey-time 2h
  ..
ipsec-policy-template ipsec1
  esp-proposal 1
    enc-alg aes128-cbc
    auth-alg hmac-sha1
    esn true
    ..
  replay-window 1024
  ..
vpn vpn-secgw
  ike-policy
    template ike1
    ..
  ipsec-policy
    template ipsec1
    ..
  local-address 10.175.0.1
  remote-address 10.225.0.1
  security-policy site-to-secgw-site
    local-ts subnet 10.100.0.64/26
    remote-ts subnet 10.250.0.192/26
    ..
  ..
..
..
system
  hostname ha2
  ..
ha
  group ha-group1
  ..
  ..
..
```

## Advanced options

### Sequence number synchronization parameters

IPsec SAs sequence numbers are regularly synchronized from the active node to the backup node. In case of switch over, this enables the new master node to take over the IPsec dataplane processing with proper sequence numbers:

For an output SA, the output sequence number<sup>1</sup> on the backup node should be greater or equal to the last sequence number used by this SA on the master node. Otherwise, the remote IPsec peer is likely to drop some IPsec packets sent by the new master until the sequence numbers comply to its replay window state.

For an input SA, the input sequence number<sup>2</sup> on the backup node should be close to the highest sequence number received on the master node. Otherwise the new master node is vulnerable to accepting replayed packets sent by an attacker, because its replay window is too late.

The pace at which sequence number synchronization is performed is configurable in the `ha seqnum-sync` sub-context:

```
ha1 running vrf main# ike ha seqnum-sync
ha1 running seqnum-sync# sync-period-time 10s
ha1 running seqnum-sync# sync-period-packets 2
ha1 running seqnum-sync# oseq-shift 65536
ha1 running seqnum-sync# / vrf main
```

- `sync-period-time` is the minimum time between two sequence number updates. An update is sent to the backup node only if the sequence number changed since last update (default 10s, 0 disables the time-based periodic update).
- `sync-period-packets` is the number of packets between two sequence number updates: if the input or output sequence number of an IPsec SA changes of at least that number since last synchronization, then an update is sent to the backup node (default 2, 0 disables the packet-based periodic update).
- `oseq-shift` is the optional IPsec SA output sequence number advance on the backup node: since sequence number cannot be synchronized in real time, the output sequence numbers on the inactive node are always late compared to the active mode. This value is added to the current output sequence number, in order to reduce or eliminate the gap between the active and the inactive node. Ideally, it should be greater or equal to the number of packets processed between two sequence number updates (default 65536).

<sup>1</sup> i.e. the record of the highest SA sequence number of a sent packet protected with this SA

<sup>2</sup> i.e. the record of the highest SA sequence number of a received packet protected with this SA

## HA-compatible virtual IP pools

IKEv1 and IKEv2 enable to assign a *virtual IP* during an IKE negotiation, i.e. an IKE initiator may request an additional IP address from the responder to use as inner IPsec tunnel address.

To proceed, the responder maintains a pool of virtual IPs (see *IKE virtual IP pools*).

If the IKE configuration makes use of virtual IP pools and HA IKE is enabled, then virtual IP leases must be synchronized between the master and the backup node.

This requires using specific HA-synchronized virtual IP pools. These pools are less flexible than standard virtual IP pools:

- address pools can only be defined as subnets, not ranges of addresses.
- no other parameters can be provided (such as a DNS, NetBIOS or DHCP server address).
- there is no state information about these pools

When enabling HA IKE, be careful of using a virtual pool defined in the `ha` context, because virtual pools defined directly in the `ike` context are not synchronized between the master and backup node.

Define the pool:

```
ha1 running vrf main# ike
ha1 running ike# ha
ha1 running ha# pool my-ha-pool address 192.168.0.0/24
ha1 running ha# ..
```

Use it in a vpn:

```
ha1 running ike# vpn vpn-secgw
ha1 running vpn vpn-secgw# vip
ha1 running vpn vpn-secgw# vip-pool my-ha-pool
ha1 running vpn vpn-secgw# ..
ha1 running ike#
```

Display the IKE configuration:

```
ha1 running ike# show config nodefault
ike
  vpn vpn-secgw
    vip-pool my-ha-pool
    (...)
  ha
    pool my-ha-pool
      address 192.168.0.0/24
      (...)
  (...)
```

**See also:**

The *IKE command reference* for details.

**VRRP**

Virtual Router Redundancy Protocol provides a way, for a set of routers, to control a virtual address and MAC address, including automatic fail-over mechanism. Such an address may be used by hosts for some service access, e.g. as static default gateway. The gain from using VRRP is a higher availability of the service without requiring automatic reconfiguration of end hosts.

The VRRP protocol is defined in RFC 3768 (VRRP v2) and RFC 5798 (VRRP v3). The major differences between VRRP v2 and VRRP v3 are the support of IPv6 and the support of low failover timer values which are only available in the latter.

- *Overview*
- *VRRP states*
- *VRRP interface settings*
  - *Version*
  - *Link interface*
  - *Priority*
  - *Init state*
  - *Virtual router identifier*
  - *Preemption*
  - *Advertisement interval and preempt delay*
  - *Gratuitous ARP delay*
  - *Trackers*
    - \* *IP address tracker*
    - \* *Fast path status tracker*
- *Synchronization of instance states*
  - *Failover groups configuration*
  - *High-availability group notification*
- *High-availability split-brain situation*
  - *Possible causes*
  - *Consequences*

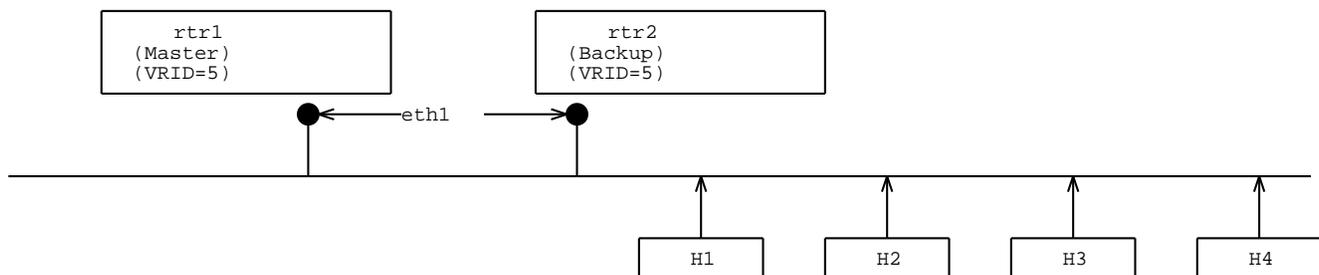
- *Solutions*
  - \* *Change the topology*
  - \* *Use a HA (High Availability) link*
  - \* *Configure a startup delay*
- *VRRP settings for virtual environments*
  - *Virtual MAC address*
  - *Unicast peering*

## Overview

There are two contexts involved in VRRP configuration:

- the `vrrp` global context, from which options common to all VRRP interfaces are set, and from which VRRP fail-over groups are defined.
- the `interface type vrrp`, from which a particular VRRP instance is configured.

Here is a simple example of VRRP, similar to *Sample Configuration 1* described in section 4.1 of RFC 3768:



The configuration of `rtr1` is done with the following commands:

```
vrouter running vrf main#
vrouter running vrf main# vrrp
vrouter running vrrp# enabled true
vrouter running vrrp# router-id vrrp_router1
vrouter running vrrp# vrrp-startup-delay 30
vrouter running vrrp# ..
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51#! link-interface eth1
vrouter running vrrp vrrp51#! vrid 5
vrouter running vrrp vrrp51# priority 200
vrouter running vrrp vrrp51# virtual-address 10.22.0.1/24
vrouter running vrrp vrrp51# commit
```

---

**Note:** The link interface *eth1* must be up and have an IP address configured.

---

Let's fetch the state after committing this configuration:

```
vrouter running vrf main# vrrp
vrouter running vrrp# show state
vrrp
  enabled true
  router-id vrrp_router1
  traps-enabled false
  ..
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51# show state
vrrp vrrp51
  vmac-xmit-base false
  preempt-delay 0
  init-state backup
  state master
  garp-delay 5
  link-interface eth1
  enabled true
  use-vmac true
  advertisement-interval 1000
  mtu 1500
  vrid 5
  oper-status UNKNOWN
  priority 200
  preempt true
  version 2
  counters
    in-errors 0
    out-discards 0
    out-octets 0
    in-octets 0
    out-unicast-pkts 9
    out-errors 0
    in-unicast-pkts 0
    in-discards 0
    ..
  ethernet
    mac-address 00:00:5e:00:01:05
    ..
  virtual-address 10.22.0.1/24
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <vrrp xmlns="urn:6wind:vrouter/vrrp">
      <enabled>true</enabled>
      <router-id>vrrp_router1</router-id>
      <traps-enabled>false</traps-enabled>
    </vrrp>
    <interface xmlns="urn:6wind:vrouter/interface">
      <vrrp xmlns="urn:6wind:vrouter/vrrp">
        <name>vrrp51</name>
        <enabled>true</enabled>
        <init-state>backup</init-state>
        <version>2</version>
        <garp-delay>5</garp-delay>
        <use-vmac>true</use-vmac>
        <vmac-xmit-base>false</vmac-xmit-base>
        <priority>200</priority>
        <preempt>true</preempt>
        <preempt-delay>0</preempt-delay>
        <advertisement-interval>1000</advertisement-interval>
        <authentication/>
        <link-interface>eth1</link-interface>
        <vrid>5</vrid>
        <virtual-address>
          <ip>10.22.0.1/24</ip>
        </virtual-address>
      </vrrp>
    </interface>
  </vrf>
</config>
```

This configuration is obtained by merging the output of the following commands:

```
vrouter running config# show config xml absolute vrf main vrrp
vrouter running config# show config xml absolute vrf main interface vrrp vrrp51
```

The configuration on the second router (backup) is similar, except the priority which should be lower than 200, and the router-id which is set to vrrp\_router2.

#### See also:

The *command reference* for details.

## VRRP states

Here are the existing VRRP states and their meanings:

- **fault** - this state is caused by a fault on a tracked element (link interface, route...). When the state switches to **fault**, the router sends a VRRP announcement with a priority of 0.
- **backup** - the router is starting or it is ready to take the virtual address in case of failure of the master.
- **master** - the only state where the router owns the virtual addresses.

## VRRP interface settings

### Version

**version** defines whether version 2 or 3 is used. Using version 3 allows faster failovers thanks to low timer values and is recommended. See *information about timers*.

```
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51# version 3
```

**Caution:** Setting IPv6 VRRP does not imply version 3. Turbo Router allows using IPv6 on VRRP version 2 although its support is deprecated.

### Link interface

**link-interface** is the interface the VRRP instance is bound to. VRRP announcements are sent from the IP address of the link interface.

On an IPv6 VRRP instances, setting an IPv6 global address on the link interface is not mandatory because the IPv6 link-local address is sufficient.

```
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51# link-interface eth1
```

## Priority

The router with the highest priority will be elected master. The other one will remain backup.

```
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51# priority 150
```

We recommend to set a priority higher than the default value of 100 on one of the routers. Otherwise, on VRRP version 2, having the same priority on both routers makes the router with the highest IP to be the master. On VRRP version 3, the first router that appears on the network is elected as master.

## Init state

`init-state` is a deprecated option that defines on which state the VRRP instance is initialized before the election occurs, at startup and after a link goes to up state.

We recommend to let it at the default value of `backup`. If you want the router election to converge quickly, use lower *timers* values.

```
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51# init-state backup
```

## Virtual router identifier

The virtual router identifier `vrid` identifies a set of VRRP routers on a LAN. Several groups of VRRP routers can coexist on the same broadcast domain as long as they all have a different `vrid`. For a given VRRP instance, the `vrid` value must be identical on both routers. We recommend to set the same `vrid` on all instances within a HA group of routers.

```
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51# vrid 30
```

**Caution:** A vMAC (virtual MAC address) is built from the `vrid` value. Setting the same `vrid` value on several groups of routers within the same broadcast domain would mean that the same vMAC is re-used at different places. See *Virtual MAC address*.

## Preemption

“To preempt” in the context of VRRP means to take over as master when the other router is master. The condition for a router to preempt is basically to receive three times a lower priority announcement than its own.

The preemption feature is enabled by default and should remain enabled. For some special cases, it can be configured using the `preempt` option:

```
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51# preempt (true|false)
```

### See also:

The *timers section* gives information about the preemption convergence time.

## Advertisement interval and preempt delay

`advertisement-interval` specifies the rate at which VRRP advertisements are sent. The configured values in our CLI are in milliseconds.

```
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51# advertisement-interval 1000
```

The available range of values depends on the VRRP version:

- On version 2, it can be configured between 1000 and 50 000 milliseconds. The configured value is rounded to the nearest multiple of 1000. For example, if it is configured with value 1400 that will be rounded to 1000, while if it is 1700 the value will be rounded to 2000.
- On version 3, it can be configured between 100 and 40 950 milliseconds. The configured value is rounded to the nearest multiple of 10. For example, if it is configured with value 2554 that will be rounded to 2550, while if it is 2555 the value will be rounded to 2560.

The advertisement interval defines the following timers:

- *Master\_Down\_Interval*: Time interval for Backup to declare Master down. Calculated as:
 
$$(3 * \text{advertisement-interval}) + \text{Skew\_Time}.$$
- *Skew\_Time*: Time to skew *Master\_Down\_Interval* in seconds. Calculated as:
  - on VRRP version 2:  $((256 - \text{local priority}) / 256).$
  - on VRRP version 3:  $((256 - \text{local priority}) * \text{advertisement-interval}) / 256$  (advertisement-interval is in seconds).
- *local priority*: Priority of the VRRP instance on the local router. A higher priority leads to a smaller *Skew\_Time*.

---

**Note:** We strongly recommend to put the same value of `advertisement-interval` on both routers.

---

The master router sends every `advertisement-interval` a VRRP announcement on all instances. If it receives a higher priority advertisement, it will switch to `backup` state after `Master_Down_Interval`.

The backup router sends no advertisement and it listens for advertisements from the other router.

- If it stops receiving advertisements, it considers that the master router might not be available anymore. It waits for `Master_Down_Interval` for confirmation before switching to `master` state.
- If it receives an advertisement from the master router with a priority of 0, it considers that the master is faulty. Similarly, it waits for `Master_Down_Interval` before switching to `master` state.
- If it receives an advertisement from the master router with a non-zero priority less than its own, it considers that it should become the master router unless the `preempt` option is disabled. It waits for `Master_Down_Interval + preempt-delay` before switching to `master` state. Default `preempt-delay` is 0.

---

**Note:** `preempt-delay` does not apply if no advertisements are received when the link interface goes up.

---

---

**Note:** A `preempt-delay` value of 0 will not allow to preempt immediately, as `Master_Down_Interval` always applies.

---

For example, if you need a failover time of 1 second, you can set:

```
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51# advertisement-interval 300
vrouter running vrrp vrrp51# preempt-delay 0
vrouter running vrrp vrrp51# priority 170
vrouter running vrrp vrrp51# version 3
```

---

**Note:** We recommend to use the same version, priority and timers on all VRRP instances within a router.

---

## Gratuitous ARP delay

A gratuitous ARP is a special type of ARP packets that broadcasts without being requested an IP address resolution to all hosts.

After transiting to `master` state, the router sends immediately several gratuitous ARP for each virtual address it owns. Then, it waits for `garp-delay` before sending a second set of gratuitous ARP. This mechanism is actually useful when the `use-vmac` option is disabled. When `use-vmac` is enabled (which is the default mode), the virtual address resolution is always the same.

---

**Note:** Since it is a delay, reducing `garp-delay` will not speed up the failover. Its default value of 5 seconds should not be changed.

---

```
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51# garp-delay 5
```

## Trackers

A tracker is used to set a VRRP instance into `fault` state when a tracking condition fails.

**Caution:** The up state and the presence of an IP address on the link interface are tracked by the VRRP instance by default. When configuring a new VRRP instance in a *failover group*, make sure the link interface matches these conditions. Otherwise, the HA group would go to `fault` state.

## IP address tracker

A VRRP instance can track IP addresses. When a tracked address is unreachable, the instance goes to `fault` state.

To enable IP tracking:

```
vrouter running config# / tracker
vrouter running tracker# icmp my-tracker vrf main address 10.100.0.1
vrouter running tracker# / vrf main
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51# track my-tracker
```

### See also:

The *Tracker guide*.

## Fast path status tracker

A VRRP instance can track the fast path status. If the fast path status does not match the configuration, the instance goes to `fault` state. This occurs for instance when the fast path is starting or stopping, or if the fast path configuration cannot be applied.

To enable fast path tracking:

```
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51# track-fast-path true
```

## Synchronization of instance states

This section explains how to configure VRRP on Turbo Router to synchronize the instance states within a VRF and to work with the HA features such as *HA contrack* and *HA IKE* .

## Failover groups configuration

A VRRP group is used to group VRRP interfaces from a given VRF that should share the same VRRP state.

The VRRP instances of a group run VRRP independently but share their state with the group. This way, all group instances are in a consistent state and follow the following rules:

- If an instance goes to `fault` state, all the instances of its group will go to `fault` state as well.
- If any backup instance stops receiving announcements from the master router, the group will not go to the master state unless all instances stop receiving announcements as well.

**Caution:** All the instances of a group must have the same priority. If they don't, it will lead to constant re-elections.

The following example shows how to group two VRRP instances:

```
vrouter running vrf main#  
vrouter running vrf main# vrrp  
vrouter running vrrp# group my-group  
vrouter running group my-group# instance vrrp51  
vrouter running group my-group# instance vrrp52  
vrouter running group my-group# commit
```

## High-availability group notification

A VRRP group or VRRP instance can control the state of a high-availability group: when the VRRP state changes, all the subscribers of the high-availability group are notified and can act accordingly.

The following example configures a VRRP instance as a controller for the high-availability group *my-ha-group*.

### See also:

*High-availability Groups* for details.

```
vrouter running config# ha group my-ha-group  
vrouter running group my-ha-group# / vrf main interface vrrp vrrp51  
vrouter running vrrp vrrp51#! link-interface eth1 vrid 5  
vrouter running vrrp vrrp51# notify-ha-group my-ha-group  
vrouter running vrrp vrrp51# commit
```

Other services support high-availability can declare themselves as subscribers for this group.

---

**Note:** If a HA group notification is configured, the following rules apply:

- all the VRRP instances of a VRF must be in a unique VRRP group. Otherwise, the configuration would not be accepted at commit time.
  - only the master router can forward traffic. It synchronizes protocol information such as neighbor and packet sequence numbers to the backup. The backup router cannot synchronize information to the master. If it receives some data plane traffic by mistake, the traffic will be dropped.
- 

### High-availability split-brain situation

A HA split-brain situation is a situation where both VRRP routers are in `master` state because a network outage stops them from exchanging VRRP announcements between them.

---

**Note:** If you are not using *HA conntrack* or *HA IKE*, HA split-brain situation will not affect Turbo Router packet processing. You can skip reading until the *Startup delay* section. Configuring `vrrp-startup-delay` has some other benefits.

---

### Possible causes

The cause of a broken communication between instances that causes a double master HA topology are the following:

- An interface may only receive traffic several seconds after the link comes up. The common causes are:
  - a switch with STP (Spanning Tree Protocol) configured in non-edge (or portfast) mode,
  - a LACP bonding coming up before its slaves interface are in LACP `collecting` state.

Since all instance interfaces are likely to face this issue at the same time at system startup, we recommend to use a `vrrp-startup-delay` of 30 seconds. To avoid an issue where all the instance interfaces would flap at the same time, it is advised as well to avoid the VRRP instance to rely on a unique logical interface or on several interfaces linked to the same switch. Consider adding a *HA link* if all the instances relies on one switch.

- There is a defect on the network on a single point of failure between all the instances. See *topology solutions*.

## Consequences

The consequences of a HA split-brain depend on the services the Turbo Router is running:

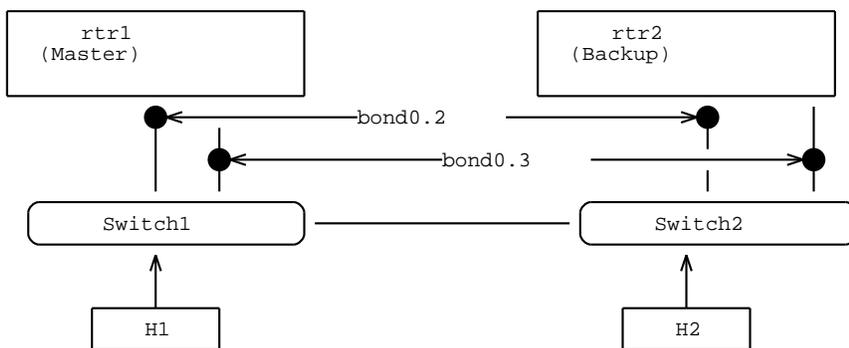
- If the nodes are using *HA conntrack* or *HA IKE* , the sequence numbers of packets are synchronized from the master to the backup or fault router. The data plane traffic is only accepted and forwarded on the master router. If the routers are all in master state, they will both forward a part of the traffic and increment their sequence numbers independently. They will not synchronize their sequence numbers one with another because a master router cannot accept synchronized information by design. As soon as the unlegitimate master router returns to the backup state, the master router will receive its traffic. However the master’s sequence number table is outdated, so this traffic will be dropped.
- Without *HA conntrack* and *HA IKE* , having several master routers in the HA topology does not affect their packet processing because the routers forward packets without checking any sequence number. The split-brain situation is the result of a fault on the network that might lead to a traffic outage but that is not caused by the routers.

## Solutions

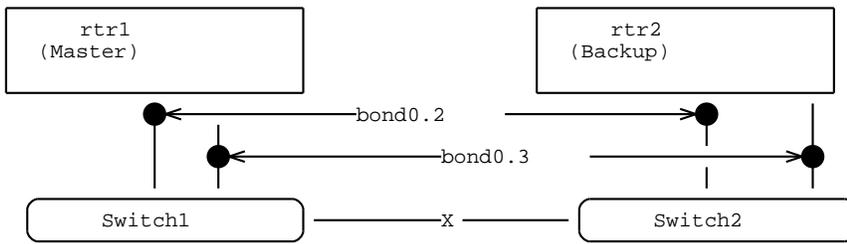
### Change the topology

We recommend to use different physical links and switches for the instances of a group to avoid this the split-brain situation.

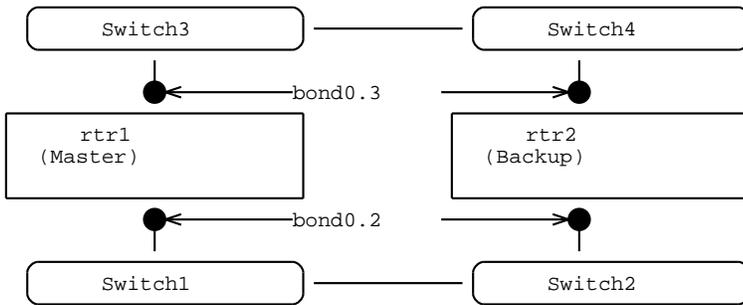
Here is a network topology with two routers running Turbo Router. Each router has a HA group containing instances on link interfaces bond0.2 and bond0.3 that are respectively vlan 2 and 3 sub-interfaces of bond0.



The link between switch1 and switch2 is now broken. The communication is totally broken between routers. As a result, both routers are master.



A solution to avoid the problem is to dispatch the VRRP instances on several switching domains to ensure that VRRP announcements for different VRRP instances take a different path. It requires adding some switches.

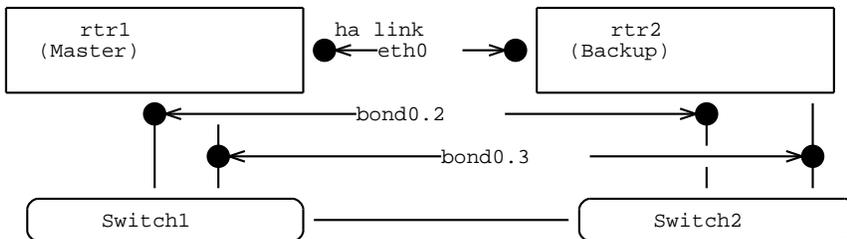


If adding some switches is not possible, consider adding a *HA link*.

### Use a HA link

The HA link is a dedicated link between two routers that is used to exchange VRRP announcements on an additional path. In case of failure on the network, when the main instances are prevented from exchanging their VRRP announcements, VRRP announcements are still exchanged on this link. The backup router knows that the other router is still up and does not switch to master state. Without the HA link, this condition would have led to have two master routers.

The following diagram shows a HA link implemented on eth0. When communication is broken between bond0 interfaces of rtr1 and rtr2, VRRP announcements are still exchanged on the HA link between rtr1 eth0 and rtr2 eth0.



To check the availability of the other router on the HA link, a VRRP instance must be bound to the link and added to the same VRRP group as the other instances.

Like any other VRRP instance, a HA link VRRP instance has an associated virtual address. It must not be used for receiving data plane traffic.

Here are some recommendations to make the HA link reliable enough:

- To be reliable enough, the HA link must not use a same path as the main interfaces.
  - If Turbo Router is running on a bare-metal server, it should be a dedicated physical link between the servers.
  - If Turbo Router is running on a virtual machine, the link should take a different physical path.
- the HA link must be bound to a physical interface, which excludes for example a bonding or a virtual interface.
- the HA interface must not be a fast path port.
- the HA link should not go through a switch if Turbo Router is running on a bare-metal server.

Also, the HA link VRRP instance:

- must use IPv4, even if the main instances are all of IPv6 type.
- must use the same VRRP version as the other instances.

We recommend to use the HA link to synchronize stateful information between the routers. `local-address` and `remote-address` would refer to the HA link ip addresses. Refer to *HA neighbor*, *HA conntrack* and *HA IKE* for details.

The `track-link-interface` option must be disabled. It prevents the VRRP HA link instance (and the instances of its group) from switching to `fault` state in case the HA link goes down.

```
vrouter running vrf main# interface vrrp eth0
vrouter running vrrp eth0# track-link-interface false
```

## Configure a startup delay

`vrrp-startup-delay` is a delay that is applied at system startup before starting to listen for VRRP advertisement. This way, no VRRP election happens before the router is ready.

We recommend to set a 30s `vrrp-startup-delay` value.

```
vrouter running vrf main#
vrouter running vrf main# vrrp
vrouter running vrrp# vrrp-startup-delay 30
```

---

**Note:** `vrrp-startup-delay` only applies at system startup but not after a link state is flapping.

---

---

**Note:** Before the `vrrp-startup-delay` timer expires, the initial state set by the `init-state` leaf in the VRRP interface configuration applies. Make sure to let it at the default backup value.

---

## VRRP settings for virtual environments

Using Turbo Router on VMs with VRRP may require to adjust the following settings:

- *Virtual MAC address*
- *Unicast peering*

---

**Note:** We recommend to keep default value of these settings if Turbo Router is running on bare-metal servers.

---

### Virtual MAC address

By default, a vMAC is associated to the virtual IP address in the host ARP and NDP neighbor table. It is also used as a source MAC address for VRRP announcements.

Some virtual network switches (eg. VMware vSwitch) are unable to deal with vMAC because they only know MAC addresses that the hypervisor has assigned to the VMs. In this case, `use-vmac` must be set to false so that the MAC address of the link interface is used instead.

```
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51# use-vmac false
```

**Caution:** When disabling `use-vmac`, make sure the ARP or NDP protocols are correctly propagated. Otherwise, some hosts might not be informed of the new master MAC address after a failover. We recommend to test some manual failovers.

- The storm control feature, if it is enabled on switches, may affect the flooding of ARP broadcasts.
- The reception of gratuitous ARP may be disabled on hosts for security reasons.

---

**Note:** Since IPv4 and IPv6 vMAC formats are different, it is not possible to define IPv4 and IPv6 virtual addresses on the same VRRP instance.

---

---

**Note:** `vmac-xmit`-based setting is only effective if `use-vmac` is true. Its default value is false. If true, the MAC address of the link interface is used for sending VRRP announcements. It should be left configured to its default value.

---

## Unicast peering

By default, the VRRP announcements are sent on multicast.

Some virtual network switches (eg. VMware vSwitch) are unable to deal with multicast MAC addresses. In this case, a `unicast-peer` must be configured to send VRRP using unicast packets instead of multicast packets. The destination IP address (192.168.222.2 in the following example) is the actual address of the same instance on the other router.

```
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51# unicast-peering 192.168.222.2
```

---

**Note:** When using values for `advertisement-interval` lower than 1s and unicast peering, we recommend to use static ARP / NDP entries to avoid latency induced by ARP / NDP requests.

---

### 3.1.11 Monitoring

#### KPIs

6WIND KPI (Key Performance Indicator) monitoring provides the ability to monitor and export Turbo Router KPIs to an [InfluxDB](https://www.influxdata.com/time-series-platform/influxdb/) (https://www.influxdata.com/time-series-platform/influxdb/) time-series database, which can then be integrated with an analytics frontend, such as [Grafana](https://grafana.com/) (https://grafana.com/). An example of InfluxDB/Grafana setup is described on 6WIND's [github](https://github.com/6WIND/supervision-grafana) (https://github.com/6WIND/supervision-grafana).

Configuring KPIs requires to:

- enable and configure the KPIs daemon to specify which KPIs to collect
- enable and configure the [Telegraf](https://www.influxdata.com/time-series-platform/telegraf) (https://www.influxdata.com/time-series-platform/telegraf) agent to export the specified KPIs to a remote InfluxDB database

To configure the KPIs daemon with everything it can collect, and the [Telegraf](https://www.influxdata.com/time-series-platform/telegraf) agent to send data to the InfluxDB server located at `http://1.1.1.1:8086`, in the `test` database, do:

```
vrouter running config# system kpi
vrouter running kpi# / vrf main kpi telegraf
vrouter running telegraf/# influxdb-output url http://1.1.1.1:8086 database test
vrouter running telegraf/# commit
```

---

**Note:** To connect [Telegraf](https://www.influxdata.com/time-series-platform/telegraf) to a secured InfluxDB instance (https URL) that is using a self-signed certificate, you must enable `insecure-skip-verify`.

---

For reference, using the deprecated way of configuring KPIs, the configuration would look like:

```

vrouter running config# vrf main kpi
vrouter running kpi/# telegraf
vrouter running telegraf/# influxdb-output url http://1.1.1.1:8086 database test
vrouter running telegraf/# commit

```

To display the state:

```

vrouter running config# show state vrf main kpi
kpi
  telegraf
    interval 10
    enabled true
    influxdb-output url http://1.1.1.1:8086 database test
  ..
..

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouter running config# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <kpi xmlns="urn:6wind:vrouter/kpi">
      <enabled>true</enabled>
      <service>fp-bridge-stats</service>
      <service>fp-context-switch-stats</service>
      <service>fp-cp-protect-stats</service>
      <service>fp-cpu-usage</service>
      <service>fp-dpvi-stats</service>
      <service>fp-ebtables-stats</service>
      <service>fp-exception-queue-stats</service>
      <service>fp-exceptions-stats</service>
      <service>fp-filling</service>
      <service>fp-filling-cg-nat</service>
      <service>fp-global-stats</service>
      <service>fp-gre-stats</service>
      <service>fp-gro-stats</service>
      <service>fp-ip-stats</service>
      <service>fp-ip6-stats</service>
      <service>fp-ipsec-stats</service>
      <service>fp-ipsec6-stats</service>
      <service>fp-cg-nat-stats</service>
      <service>fp-ports-stats</service>
      <service>fp-status</service>
      <service>fp-vlan-stats</service>
      <service>fp-vxlan-stats</service>
    </kpi>
  </system>
</config>

```

(continues on next page)

(continued from previous page)

```

<service>network-nic-eth-stats</service>
<service>network-nic-hw-info</service>
<service>network-nic-traffic-stats</service>
<service>product-license</service>
<service>product-version</service>
<service>system-cpu-usage</service>
<service>system-disk-usage</service>
<service>system-memory</service>
<service>system-numa-stats</service>
<service>system-processes</service>
<service>system-soft-interrupts-stats</service>
<service>system-uptime</service>
<service>system-user-count</service>
<service>system-users</service>
</system>
<vrf>
  <name>main</name>
  <kpi xmlns="urn:6wind:vrouter/kpi">
    <telegraf xmlns="urn:6wind:vrouter/kpi/telegraf">
      <enabled>true</enabled>
      <interval>10</interval>
      <influxdb-output>
        <url>http://1.1.1.1:8086</url>
        <database>test</database>
      </influxdb-output>
    </telegraf>
    <interface xmlns="urn:6wind:vrouter/interface"/>
  </kpi>
</vrf>
</config>

```

## sFlow

sFlow is a technology for monitoring traffic in data networks containing switches and routers. It consists of an sFlow Agent running on the router, and a central sFlow Collector.

The sFlow Agent uses sampling technology to capture traffic statistics from the device it is monitoring. sFlow Datagrams are used to immediately forward the sampled traffic statistics to an sFlow Collector for analysis.

More information is available in RFC 3176 and [sflow.org](http://sflow.org).

To configure sFlow you need to specify the collector endpoint and which interfaces will be polled.

For each interface, you can tune the sampling interval or let the system choose a value according to the speed of interface.

Configuration example:

```
vrouters running config# vrf main sflow
vrouters running sflow# sflow-collector 10.0.0.3 port 6343
vrouters running sflow# sflow-interface eth1
vrouters running sflow# sflow-sampling speed 10G rate auto
vrouters running sflow# commit
```

To display the sFlow state:

```
vrouters running config# show state vrf main sflow
sflow
  sflow-collector 10.0.0.3 port 6343
  enabled true
  sflow-interface eth1
  sflow-port 36343
  polling disabled
  sflow-sampling speed other rate auto
  sflow-sampling speed 100M rate auto
  sflow-sampling speed 1G rate auto
  sflow-sampling speed 10G rate auto
  sflow-sampling speed 40G rate auto
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouters> show config xml absolute vrf main sflow
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <sflow xmlns="urn:6wind:vrouter/sflow">
      <enabled>true</enabled>
      <polling>disabled</polling>
      <sflow-port>36343</sflow-port>
      <sflow-collector>
        <address>10.0.0.3</address>
        <port>6343</port>
      </sflow-collector>
      <sflow-interface>
        <name>eth1</name>
      </sflow-interface>
      <sflow-sampling>
        <speed>10G</speed>
        <rate>auto</rate>
      </sflow-sampling>
    </sflow>
  </vrf>
</config>
```

(continues on next page)

(continued from previous page)

```
</vrf>
</config>
```

**See also:**

The *command reference* for details.

**SNMP****Overview**

Simple Network Management Protocol (SNMP) is an Internet Standard protocol for collecting and organizing information about managed devices on IP networks and for modifying that information to change device behavior. It is specified in **RFC 1157** (<https://tools.ietf.org/html/rfc1157.html>).

Configuration example:

```
vrouter running config# vrf main snmp
vrouter running snmp# static-info contact oam@my-company.com
vrouter running snmp# static-info location "Santa Barbara"
vrouter running snmp# community public authorization read-only source 10.0.0.0/24
vrouter running snmp# traps destination 10.0.0.200 notification-type TRAP2 community_
↳public
vrouter running snmp# traps process-check
vrouter running snmp# traps link-status-check
vrouter running snmp# traps load-check threshold 95
vrouter running snmp# /
vrouter running config# commit
Configuration committed.
```

To display the current SNMP state:

```
vrouter running config# show state vrf main snmp
snmp
  enabled true
  static-info
    location "Copacabana, Rio de Janeiro"
    contact oam@my-company.com
    ..
  community public
    source 10.0.0.0/24
    authorization read-only
    ..
  traps
```

(continues on next page)

(continued from previous page)

```

destination 10.0.0.200 community public port 162 notification-type TRAP2
link-status-check enabled true frequency 60s
process-check enabled true frequency 2s
load-check enabled true threshold 95
..
..

```

The same configuration can be made using this NETCONF XML request:

```

vrouters running config# show config xml absolute vrf main snmp
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <snmp xmlns="urn:6wind:vrouter/snmp">
      <static-info>
        <contact>oam@my-company.com</contact>
        <location>Copacabana, Rio de Janeiro</location>
      </static-info>
      <traps>
        <destination>
          <host>10.0.0.200</host>
          <notification-type>TRAP2</notification-type>
          <community>public</community>
          <port>162</port>
        </destination>
        <process-check>
          <frequency>2s</frequency>
          <enabled>true</enabled>
        </process-check>
        <link-status-check>
          <frequency>60s</frequency>
          <enabled>true</enabled>
        </link-status-check>
        <load-check>
          <threshold>95</threshold>
          <enabled>true</enabled>
        </load-check>
      </traps>
      <community>
        <name>public</name>
        <authorization>read-only</authorization>
        <source>10.0.0.0/24</source>
      </community>
    </snmp>
  </vrf>
</config>

```

(continues on next page)

(continued from previous page)

```
</vrf>
</config>
```

## Monitoring several VRFs

### Configuration

Monitoring different VRFs relies on a master VRF that provides access to the management network. Slave VRFs (that don't have access to the management network) are linked to the master VRF.

To link a slave VRF:

- add it to the list of monitored VRFs in the SNMP context of the master VRF,
- configure an identifier for the slave VRF, which will act as:
  - a community for SNMP v1, v2c and traps,
  - a context for SNMP v3.

Here is a configuration example to link VRF vrf1 with identifier vrf1 to VRF main:

```
vrouter running config# vrf vrf1 snmp enabled false
vrouter running config# vrf main snmp
vrouter running snmp# static-info contact oam@my-company.com
vrouter running snmp# static-info location "Santa Barbara"
vrouter running snmp# community public authorization read-only source 10.0.0.0/24
vrouter running snmp# access-control user alice auth-password Password
vrouter running snmp# access-control group admin user alice authorization read-write_
↳security-level auth
vrouter running snmp# traps destination 10.0.0.200 notification-type TRAP2 community_
↳public
vrouter running snmp# traps process-check
vrouter running snmp# traps link-status-check
vrouter running snmp# traps load-check threshold 95
vrouter running snmp# monitored-vrf vrf1
vrouter running monitored-vrf vrf1# identifier vrf1 authorization read-only source 10.
↳0.0.0/24
vrouter running monitored-vrf vrf1# traps destination 10.0.0.200 community vrf1
vrouter running monitored-vrf vrf1# commit
Configuration committed.
```

### Note:

- When a VRF is linked to another one, its SNMP context must be disabled.

- For SNMP v3, both user and identifier authorizations are applied. Therefore, the most restrictive one is taken into account (in the example above: read-only).

The same configuration can be made using this NETCONF XML request:

```
vrouters running config# show config xml absolute vrf slave snmp
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>slave</name>
    <snmp xmlns="urn:6wind:vrouter/snmp">
      <enabled>>false</enabled>
      <static-info/>
      <access-control/>
      <traps/>
    </snmp>
  </vrf>
</config>
vrouters running config# show config xml absolute vrf main snmp
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <snmp xmlns="urn:6wind:vrouter/snmp">
      <enabled>>true</enabled>
      <static-info>
        <contact>oam@my-company.com</contact>
        <location>Santa Barbara</location>
      </static-info>
      <access-control>
        <user>
          <name>alice</name>
          <auth-password>Password</auth-password>
          <auth-method>sha</auth-method>
          <priv-protocol>aes</priv-protocol>
        </user>
        <group>
          <name>admin</name>
          <user>alice</user>
          <authorization>read-write</authorization>
          <security-level>auth</security-level>
        </group>
      </access-control>
      <traps>
        <destination>
          <host>10.0.0.200</host>
          <notification-type>TRAP2</notification-type>
        </destination>
      </traps>
    </snmp>
  </vrf>
</config>
```

(continues on next page)

(continued from previous page)

```
    <community>public</community>
    <port>162</port>
    <protocol>udp</protocol>
</destination>
<process-check>
  <frequency>2s</frequency>
  <enabled>>true</enabled>
</process-check>
<link-status-check>
  <frequency>60s</frequency>
  <enabled>>true</enabled>
</link-status-check>
<load-check>
  <threshold>95</threshold>
  <enabled>>true</enabled>
</load-check>
</traps>
<community>
  <name>public</name>
  <authorization>read-only</authorization>
  <source>10.0.0.0/24</source>
</community>
<monitored-vrf>
  <name>slave</name>
  <traps>
    <destination>
      <host>10.0.0.200</host>
      <community>private</community>
    </destination>
  </traps>
  <identifier>
    <name>private</name>
    <authorization>read-only</authorization>
    <source>10.0.0.0/24</source>
  </identifier>
</monitored-vrf>
</snmp>
</vrf>
</config>
```

## SNMP requests

Then SNMP information of the slave VRFs can be retrieved through:

- the community parameter for SNMP v1 and v2,
- the context name for SNMP v3.

Here are some SNMP request examples with the configuration example above, assuming that:

- the SNMP server is listening on 10.0.0.2,
- the main VRF has 3 interfaces, the second one being called mgmt,
- the vrf1 VRF has 2 interfaces, the second one being called eth\_out.

SNMP requests on main VRF:

```
# snmpwalk -Ovq -v 2c -c public 10.0.0.2 IF-MIB::ifIndex
1
2
3

# snmpget -Ovq -v 2c -c public 10.0.0.2 IF-MIB::ifName.2
mgmt

# snmpwalk -Ovq -v 3 -l authNoPriv -u alice -a MD5 -A "Password" 10.0.0.2 IF-
↳MIB::ifIndex
1
2
3

# snmpget -Ovq -v 3 -l authNoPriv -u alice -a MD5 -A "Password" IF-MIB::ifName.2
mgmt
```

SNMP requests on vrf1 VRF:

```
# snmpwalk -Ovq -v 2c -c vrf1 10.0.0.2 IF-MIB::ifIndex
1
2

# snmpget -Ovq -v 2c -c vrf1 10.0.0.2 IF-MIB::ifName.2
eth_out

# snmpwalk -Ovq -v 3 -n vrf1 -l authNoPriv -u alice -a MD5 -A "Password" IF-
↳MIB::ifIndex
1
2
```

(continues on next page)

(continued from previous page)

```
# snmpget -Ovq -v 3 -n vrf1 -l authNoPriv -u alice -a MD5 -A "Password" IF-MIB::ifName.
↳2
eth_out
```

## SNMP traps

The SNMP traps from the different VRFs can be distinguished thanks to the community name.

Here is a `snmptrapd` configuration example, using the `public` community for trap events from the main VRF and the `vrf1` community for the `vrf1` VRF.

`/etc/snmp/snmptrapd.conf`:

```
# Log trap from public and vrf1 communities
authCommunity log public
authCommunity log vrf1
# override default format for TRAP2 to display the community information (%P option)
format print2 %.4y-%.2m-%.2l %.2h:%.2j:%.2k %P %B [%b]:\n%\v\n
```

Start the `snmptrapd` daemon and log the trap events into `/tmp/snmptrapd.log`.

```
# snmptrapd -Lf /tmp/snmptrapd.log
```

Here are some log examples when interfaces get down in the `vrf1` and main VRFs.

```
# cat /tmp/snmptrapd.log
2019-12-10 10:38:44 TRAP2, SNMP v2c, community vrf1 <UNKNOWN> [UDP: [10.0.0.2]:42616->
↳[10.0.0.200]:162]:
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (8205) 0:01:22.05      SNMPv2-
↳MIB::snmpTrapOID.0 = OID: IF-MIB::linkDown      IF-MIB::ifIndex.2 = INTEGER: 2      ↳
↳IF-MIB::ifAdminStatus.2 = INTEGER: down(2)      IF-MIB::ifOperStatus.2 = INTEGER:↳
↳down(2)      SNMPv2-MIB::snmpTrapEnterprise.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.
↳10
2019-12-10 10:41:04 TRAP2, SNMP v2c, community public <UNKNOWN> [UDP: [10.0.0.2]:51157-
↳>[10.0.0.200]:162]:
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (22211) 0:03:42.11      SNMPv2-
↳MIB::snmpTrapOID.0 = OID: IF-MIB::linkDown      IF-MIB::ifIndex.2 = INTEGER: 2      ↳
↳IF-MIB::ifAdminStatus.2 = INTEGER: down(2)      IF-MIB::ifOperStatus.2 = INTEGER:↳
↳down(2)      SNMPv2-MIB::snmpTrapEnterprise.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.
↳10
```

### See also:

The *SNMP command reference* for details.

## **Supported mibs**

### **Standard MIBs support**

Turbo Router supports the following MIBs (Management Information Bases):

| MIB name                | Reference   | Root OID  |
|-------------------------|---|---|
| DISMAN-EVENT-MIB        | <b>RFC</b> <b>2981</b><br>( <a href="https://tools.ietf.org/html/rfc2981.html">https://tools.ietf.org/html/rfc2981.html</a> ) | .iso.org.dod.internet.mgmt.mib-2.<br>dismanEventMIB<br>.1.3.6.1.2.1.88              |
| EtherLike-MIB           | <b>RFC</b> <b>3635</b><br>( <a href="https://tools.ietf.org/html/rfc3635.html">https://tools.ietf.org/html/rfc3635.html</a> ) | .iso.org.dod.internet.mgmt.mib-2.<br>etherMIB<br>.1.3.6.1.2.1.35                    |
| IF-MIB                  | <b>RFC</b> <b>2863</b><br>( <a href="https://tools.ietf.org/html/rfc2863.html">https://tools.ietf.org/html/rfc2863.html</a> ) | .iso.org.dod.internet.mgmt.mib-2.<br>interfaces<br>.1.3.6.1.2.1.2                   |
| IF-MIB                  | <b>RFC</b> <b>2863</b><br>( <a href="https://tools.ietf.org/html/rfc2863.html">https://tools.ietf.org/html/rfc2863.html</a> ) | .iso.org.dod.internet.mgmt.mib-2.ifMIB<br>1.3.6.1.2.1.31                            |
| IP-MIB                  | <b>RFC</b> <b>4293</b><br>( <a href="https://tools.ietf.org/html/rfc4293.html">https://tools.ietf.org/html/rfc4293.html</a> ) | .iso.org.dod.internet.mgmt.mib-2.ip<br>1.3.6.1.2.1.4                                |
| IP-MIB                  | <b>RFC</b> <b>4293</b><br>( <a href="https://tools.ietf.org/html/rfc4293.html">https://tools.ietf.org/html/rfc4293.html</a> ) | .iso.org.dod.internet.mgmt.mib-2.icmp<br>1.3.6.1.2.1.5                              |
| IP-FORWARD-MIB          | <b>RFC</b> <b>4292</b><br>( <a href="https://tools.ietf.org/html/rfc4292.html">https://tools.ietf.org/html/rfc4292.html</a> ) | .iso.org.dod.internet.mgmt.mib-2.ip.<br>ipForward<br>.1.3.6.1.2.1.4.24              |
| IPV6-MIB                | <b>RFC</b> <b>2465</b><br>( <a href="https://tools.ietf.org/html/rfc2465.html">https://tools.ietf.org/html/rfc2465.html</a> ) | .iso.org.dod.internet.mgmt.mib-2.<br>ipv6MIB<br>.1.3.6.1.2.1.55                     |
| HOST-RESOURCES-MIB      | <b>RFC</b> <b>2790</b><br>( <a href="https://tools.ietf.org/html/rfc2790.html">https://tools.ietf.org/html/rfc2790.html</a> ) | .iso.org.dod.internet.mgmt.mib-2.host<br>1.3.6.1.2.1.25                             |
| RFC1213-MIB             | <b>RFC</b> <b>1213</b><br>( <a href="https://tools.ietf.org/html/rfc1213.html">https://tools.ietf.org/html/rfc1213.html</a> ) | .iso.org.dod.internet.mgmt.mib-2<br>1.3.6.1.2.1                                     |
| SNMP-FRAMEWORK-MIB      | <b>RFC</b> <b>3411</b><br>( <a href="https://tools.ietf.org/html/rfc3411.html">https://tools.ietf.org/html/rfc3411.html</a> ) | .iso.org.dod.internet.snmpV2.<br>snmpModules.snmpFrameworkMIB<br>.1.3.6.1.6.3.10    |
| SNMP-MPD-MIB            | <b>RFC</b> <b>3412</b><br>( <a href="https://tools.ietf.org/html/rfc3412.html">https://tools.ietf.org/html/rfc3412.html</a> ) | .iso.org.dod.internet.snmpV2.<br>snmpModules.snmpMPDMIB<br>.1.3.6.1.6.3.11.         |
| SNMP-NOTIFICATION-MIB   | <b>RFC</b> <b>3413</b><br>( <a href="https://tools.ietf.org/html/rfc3413.html">https://tools.ietf.org/html/rfc3413.html</a> ) | .iso.org.dod.internet.snmpV2.<br>snmpModules.snmpNotificationMIB<br>.1.3.6.1.6.3.13 |
| SNMP-TARGET-MIB         | <b>RFC</b> <b>3413</b><br>( <a href="https://tools.ietf.org/html/rfc3413.html">https://tools.ietf.org/html/rfc3413.html</a> ) | .iso.org.dod.internet.snmpV2.<br>snmpModules.snmpTargetMIB<br>.1.3.6.1.6.3.12       |
| SNMP-USER-BASED-SM-MIB  | <b>RFC</b> <b>3414</b><br>( <a href="https://tools.ietf.org/html/rfc3414.html">https://tools.ietf.org/html/rfc3414.html</a> ) | .iso.org.dod.internet.snmpV2.<br>snmpModules.snmpUsmMIB<br>.1.3.6.1.6.3.15          |
| SNMP-VIEW-BASED-ACM-MIB | <b>RFC</b> <b>3415</b><br>( <a href="https://tools.ietf.org/html/rfc3415.html">https://tools.ietf.org/html/rfc3415.html</a> ) | .iso.org.dod.internet.snmpV2.<br>snmpModules.snmpVacmMIB<br>.1.3.6.1.6.3.16         |
| SNMPv2-MIB              | <b>RFC</b> <b>1213</b><br>( <a href="https://tools.ietf.org/html/rfc1213.html">https://tools.ietf.org/html/rfc1213.html</a> ) | .iso.org.dod.internet.mgmt.mib-2.system<br>1.3.6.1.2.1.1                            |
| SNMPv2-MIB              | <b>RFC</b> <b>1213</b><br>( <a href="https://tools.ietf.org/html/rfc1213.html">https://tools.ietf.org/html/rfc1213.html</a> ) | .iso.org.dod.internet.mgmt.mib-2.snmp<br>1.3.6.1.2.1.11                             |
| SNMPv2-MIB              | <b>RFC</b> <b>3418</b><br>( <a href="https://tools.ietf.org/html/rfc3418.html">https://tools.ietf.org/html/rfc3418.html</a> ) | .iso.org.dod.internet.snmpV2.<br>snmpModules.snmpMIB                                |

## Interface Alias

IF-MIB::ifAlias parameter reflects the value of Linux network device aliases.

Interface aliases set through the CLI, iproute2 or sysfs can be read through SNMP in this parameter.

## Control Plane Routing

Supported MIBs:

| MIB name                | Reference  | Root OID   |
|-------------------------|--|--|
| BGP4-MIB                |  | .iso.org.dod.internet.mgmt.mib-2.bgp<br>.1.3.6.1.2.1.15                                      |
| GNOME-PRODUCT-ZEBRA-MIB |  | .iso.org.dod.internet.private.enterprises.gnome.gnomeProducts.zebra<br>.1.3.6.1.4.1.3319.1.2 |
| OSPF-MIB                |  | .iso.org.dod.internet.mgmt.mib-2.ospf<br>.1.3.6.1.2.1.14                                     |
| OSPF-TRAP-MIB           |  | .iso.org.dod.internet.mgmt.mib-2.ospf.ospfTrap<br>.1.3.6.1.2.1.14.16                         |
| OSPFv3-MIB              | <b>RFC 5643</b><br>( <a href="https://tools.ietf.org/html/rfc5643">https://tools.ietf.org/html/rfc5643</a> ) | .iso.org.dod.internet.mgmt.mib-2.ospfv3MIB<br>.2.1.191                                       |
| RIPv2-MIB               |  | .iso.org.dod.internet.mgmt.mib-2.rip2<br>.1.3.6.1.2.1.23                                     |

## HA VRRP

Supported MIBs:

| MIB name       | Reference  | Root OID   |
|----------------|--|--|
| KEEPALIVED-MIB |  | .iso.org.dod.internet.private.enterprises.debian.project.Keepalived<br>.1.3.6.1.4.1.9586.100.5 |
| VRRP-MIB       | <b>RFC 2787</b><br>( <a href="https://tools.ietf.org/html/rfc2787">https://tools.ietf.org/html/rfc2787</a> ) | .iso.org.dod.internet.mgmt.mib-2.vrrpMIB<br>.1.2.1.68  |
| VRRPv3-MIB     | <b>RFC 6527</b><br>( <a href="https://tools.ietf.org/html/rfc6527">https://tools.ietf.org/html/rfc6527</a> ) | .iso.org.dod.internet.mgmt.mib-2.vrrpv3MIB<br>.1.2.1.207                                       |

## IPsec/IKE

- IPsec monitoring
- IKE monitoring

Supported MIB:

| MIB name           | Refer-ence | Root OID  |
|--------------------|------------|---|
| SW-6WIND-IPSEC-MIB | 6WIND      | .iso.org.dod.internet.private.enterprises.sw6wind.swSecurity.swIpssecMib<br>.1.3.6.1.4.1.7336.2.1 |

## 3.1.12 Services

### LLDP

802.1AB Link-Layer Discovery Protocol (LLDP) provides information to devices that are directly adjacent to them on the local LAN.

LLDP sends information periodically and at link status change time to indicate the configuration parameters of the device.

This protocol allows the router to:

- advertise its identity and capabilities on the local network
- receive the same information from a physically adjacent layer 2 peer

LLDP uses Ethernet as its transport protocol, the Ethernet type for LLDP is 0x88CC.

User can control which information is being sent from the router:

- description of the device
- system name
- the IP address to reach the device on LLDP port

User has to define the list of interfaces on which LLDP is active.

The chassis ID will be set automatically.

To configure LLDP to start on the eth0 interface, reachable on the 10.0.0.1 address, with name vrouter and description Router, do:

```
vrouter running config# vrf main
vrouter running vrf main# lldp
vrouter running lldp# enabled true
```

(continues on next page)

(continued from previous page)

```
vrouter running lldp# interface eth0
vrouter running interface eth0# ..
vrouter running lldp# system-name vrouter
vrouter running lldp# system-description Router
vrouter running lldp# management-address 10.0.0.1
vrouter running lldp# commit
Configuration applied.
```

To display the LLDP state:

```
vrouter running config# show state vrf main lldp
lldp
  management-address 10.0.0.1
  system-name vrouter
  system-description Router
  chassis-id-type mac-address
  counters
    tlv-discard 0
    frame-in 0
    frame-out 0
    frame-discard 0
  ..
  enabled true
  chassis-id de:ad:de:01:02:03
  hello-timer 30
  interface eth0
    enabled true
    counters
      frame-out 1
      frame-discard 0
      tlv-discard 0
      frame-in 0
    ..
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter> show config xml absolute vrf main lldp
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <lldp xmlns="urn:6wind:vrouter/lldp">
      <management-address>10.0.0.1</management-address>
```

(continues on next page)

(continued from previous page)

```
<system-name>vrouter</system-name>
<system-description>Router</system-description>
<enabled>>true</enabled>
<interface>
  <name>eth0</name>
  <enabled>>true</enabled>
</interface>
</lldp>
</vrf>
</config>
```

**See also:**

The *command reference* for details.

**DHCP server**

- *Overview*
- *Subnet configuration*
  - *Address range management*
  - *Subnet interface*
  - *Default gateway*
  - *Host management and static address assignment*
- *DHCP options*
  - *Domain name*
  - *DNS server address*
  - *NetBIOS name server*
  - *NetBIOS node type*
  - *Lease lengths management*
- *Configuration example*
- *Displaying DHCP server leases*

## Overview

A DHCP server is typically used to configure the IPv4 addresses of the hosts connected to its different LAN subnets, upon their request.

It needs at least one IPv4 subnet on which one interface is configured and a range of addresses to be allocated to DHCP clients.

You can configure the DHCP server in the `dhcp server` context:

```
vrouter running config# vrf VRFNAME dhcp server
```

**VRFNAME** VRF name on which the DHCP server must run.

### See also:

The *DHCP server command reference* for details.

## Subnet configuration

- Specify which subnet the DHCP server should serve:

```
vrouter running server# subnet A.B.C.D/M
```

**A.B.C.D/M** IPv4 subnet address with prefix mask length.

Each subnet has its configuration parameters in a subcontext.

## Address range management

The DHCP server manages a range of addresses to be allocated to DHCP clients.

- Specify a range of addresses in a defined subnet:

```
vrouter running subnet A.B.C.D/M# range A1.B1.C1.D1 A2.B2.C2.D2
```

**A1.B1.C1.D1** First address of the range.

**A2.B2.C2.D2** Last address of the range.

---

**Note:** The A1.B1.C1.D1 – A1.B2.C2.D2 address range must be included in the configured subnet.

---

## Subnet interface

- Specify on which interface the server should listen to DHCP requests:

```
vrouter running subnet A.B.C.D/M# interface IFNAME
```

**IFNAME** The interface name.

---

### Note:

- The interface must be able to receive broadcast packets.
  - If this option is not set, the DHCP server will use the first interface that has an IP address corresponding to the subnet.
- 

## Default gateway

- Specify the list of default gateways to provide to DHCP clients, by order of preference:

```
vrouter running subnet A.B.C.D/M# default-gateway A.B.C.D
```

**A.B.C.D** IPv4 address of the default gateway provided to the hosts.

## Host management and static address assignment

You can reserve a specific IPv4 address for a given host (mainly servers, whose IPv4 address is supposed to be stable).

- Define a fixed IPv4 address for a host:

```
vrouter running subnet A.B.C.D/M# host NAME mac-address XX:XX:XX:XX:XX:XX ip-  
->address A.B.C.D
```

**NAME** The DHCP client host name.

**XX:XX:XX:XX:XX:XX** Ethernet MAC address (e.g 00:02:b3:39:ba:d2).

**A.B.C.D** IPv4 address to be provided to the host.

---

**Note:** Like ranges, the host IP address must be included in the configured subnet.

---

**Warning:** Static address assignments are not written in the machine leases, only the dynamic addresses are. As a consequence, they are neither displayed in `show dhcp-server` nor in `show state`.

## DHCP options

DHCP options can be specified in the root DHCP server context or overwritten per subnet.

### Domain name

- Specify the domain name to use by default:

```
vrouter running subnet A.B.C.D/M# dhcp-options domain-name NAME
```

**NAME** Domain name to send to clients.

### DNS server address

- Specify the list of default DNS servers, by order of preference:

```
vrouter running subnet A.B.C.D/M# dhcp-options domain-name-server A.B.C.D
```

**A.B.C.D** IPv4 server address provided to the hosts.

### NetBIOS name server

- Specify the list of default NetBIOS/WINS servers, by order of preference:

```
vrouter running subnet A.B.C.D/M# dhcp-options netbios-name-server A.B.C.D
```

**A.B.C.D** IPv4 NetBIOS server address provided to the hosts.

### NetBIOS node type

- Specify the NetBIOS node as broadcast mode and ignore WINS server address:

```
vrouter running subnet A.B.C.D/M# dhcp-options netbios-node-type b-mode
```

- Specify the NetBIOS node as always point-to-point and never broadcast request:

```
vrouter running subnet A.B.C.D/M# dhcp-options netbios-node-type p-mode
```

- Specify the NetBIOS node as `try broadcast first` if it fails to use `WINS address`:

```
vrouter running subnet A.B.C.D/M# dhcp-options netbios-node-type m-mode
```

- Specify the NetBIOS node as `hybrid mode` (starts with `WINS address`, then use `broadcast request`).

```
vrouter running subnet A.B.C.D/M# dhcp-options netbios-node-type h-mode
```

## Lease lengths management

You can define the lease lengths for the DHCP clients.

- Define the default lease length:

```
vrouter running subnet A.B.C.D/M# default-lease-time VALUE
```

- Define the maximum lease length:

```
vrouter running subnet A.B.C.D/M# max-lease-time VALUE
```

**VALUE** Lease length in seconds (between 180 and 31536000 included).

## Configuration example

```
vrouter> edit running
vrouter running config# vrf main
vrouter running vrf main# dhcp
vrouter running dhcp# server
vrouter running server# enabled true
vrouter running server# dhcp-options
vrouter running dhcp-options# domain-name-server 10.0.0.1
vrouter running dhcp-options# domain-name-server 10.0.0.2
vrouter running dhcp-options# ..
vrouter running server# subnet 1.0.0.0/24
vrouter running subnet 1.0.0.0/24# range 1.0.0.1 1.0.0.50
vrouter running subnet 1.0.0.0/24# range 1.0.0.51 1.0.0.100
vrouter running subnet 1.0.0.0/24# ..
vrouter running server# subnet 2.0.0.0/24
vrouter running subnet 2.0.0.0/24# interface eth0
vrouter running subnet 2.0.0.0/24# range 2.0.0.1 2.0.0.100
vrouter running subnet 2.0.0.0/24# ..
vrouter running server# ..
vrouter running dhcp# ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter> show config xml absolute vrf main dhcp server
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <dhcp xmlns="urn:6wind:vrouter/dhcp">
      <server>
        <enabled>true</enabled>
        <default-lease-time>43200</default-lease-time>
        <max-lease-time>86400</max-lease-time>
        <dhcp-options>
          <domain-name-server>10.0.0.1</domain-name-server>
          <domain-name-server>10.0.0.2</domain-name-server>
        </dhcp-options>
        <subnet>
          <prefix>1.0.0.0/24</prefix>
          <dhcp-options/>
          <range>
            <start-ip>1.0.0.1</start-ip>
            <end-ip>1.0.0.50</end-ip>
          </range>
          <range>
            <start-ip>1.0.0.51</start-ip>
            <end-ip>1.0.0.100</end-ip>
          </range>
        </subnet>
        <subnet>
          <prefix>2.0.0.0/24</prefix>
          <dhcp-options/>
          <interface>eth0</interface>
          <range>
            <start-ip>2.0.0.1</start-ip>
            <end-ip>2.0.0.100</end-ip>
          </range>
        </subnet>
      </server>
    </dhcp>
  </vrf>
</config>
```

## Displaying DHCP server leases

- Display the DHCP server's leases:

```
vrouter> show dhcp-server
```

- Display the DHCP server's leases in a specific VRF:

```
vrouter> show dhcp-server vrf VRFNAME
```

**VRFNAME** The VRF name.

## Example

```
vrouter> show dhcp-server vrf vrf1
authoring-byte-order little-endian;

server-duid "\000\001\000\001#\310\357\010\336\355\001\320\220\235";

lease 10.100.0.3 {
  starts 3 2019/01/09 17:42:36;
  ends 3 2019/01/09 18:42:36;
  cltt 3 2019/01/09 17:42:36;
  binding state active;
  next binding state free;
  rewind binding state free;
  hardware ethernet de:ed:01:e4:13:29;
}
```

## DHCP relay

- *Overview*
- *DHCP relay configuration*
- *DHCP configuration options*
  - *Handle option*
  - *Drop unmatched packets*
  - *Maximum hop*
  - *Maximum packet size*
- *Configuration example*

## Overview

The DHCP relay listens for DHCP queries and responses. When a query is received from a client, it is forwarded to the specified DHCP server(s). When a reply is received from a server, it is forwarded to the client that made the initial request.

The DHCP relay needs at least the IP address of a reachable DHCP server.

You can configure the DHCP relay parameters in the `dhcp relay` context.

```
vrouter running config# vrf VRFNAME dhcp relay
```

**VRFNAME** VRF name on which the DHCP relay must run.

### See also:

The *DHCP relay command reference* for details.

## DHCP relay configuration

- To relay the DHCP clients' requests to DHCP servers, the DHCP relay must know the IPv4 address of a DHCP server.

```
vrouter running relay# dhcp-server A.B.C.D
```

**A.B.C.D** IPv4 address of a DHCP server connected to the DHCP relay.

- By default the DHCP relay will listen on all broadcast interfaces. But it's also possible to specify one or more interfaces on which listen:

```
vrouter running dhcp-server A.B.C.D# interface IFNAME
```

**IFNAME** Name of an interface to which a DHCP server is connected.

- The `dhcp-server` configuration can be disabled:

```
vrouter running dhcp-server A.B.C.D# enabled false
```

## DHCP configuration options

DHCP relay options can be specified in the root DHCP relay context or overwritten per `dhcp-server`.

## Handle option

- Specify the handling policy of DHCPv4 packets that already contain relay agent options:

```
vrouter running dhcp-server A.B.C.D# handle-option append|replace|forward|discard
```

**append** Append its own set of relay options to the packet.

**replace** Replace the existing agent option field.

**forward** Forward the packet unchanged.

**discard** Discard the packet.

## Drop unmatched packets

- Packets coming from upstream servers who contains relay agent information options that indicate they were generated in response to a query that came via a different relay agent can be dropped:

```
vrouter running dhcp-server A.B.C.D# drop-unmatched true
```

## Maximum hop

- Specify the maximum hop count before discard a packet:

```
vrouter running dhcp-server A.B.C.D# hop-count <0-255>
```

## Maximum packet size

- Specify the maximum packet size to send to a DHCPv4 server. If a DHCP packet size surpasses this value it will be forwarded without appending relay agent information:

```
vrouter running dhcp-server A.B.C.D# max-size <64-1400>
```

## Configuration example

```
vrouter> edit running
vrouter running config# vrf main
vrouter running vrf main# dhcp
vrouter running dhcp# relay
vrouter running relay# hop-count 5
vrouter running relay# dhcp-server 1.0.0.1
```

(continues on next page)

(continued from previous page)

```

vrouters running dhcp-server 1.0.0.1# interface eth0
vrouters running dhcp-server 1.0.0.1# interface eth1
vrouters running dhcp-server 1.0.0.1# drop-unmatched true
vrouters running dhcp-server 1.0.0.1# ..
vrouters running relay# ..
vrouters running dhcp#

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouters> show config xml absolute vrf main dhcp relay
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <dhcp xmlns="urn:6wind:vrouter/dhcp">
      <relay>
        <enabled>true</enabled>
        <handle-option>append</handle-option>
        <drop-unmatched>false</drop-unmatched>
        <hop-count>5</hop-count>
        <max-size>576</max-size>
        <dhcp-server>
          <address>1.0.0.1</address>
          <enabled>true</enabled>
          <interface>eth0</interface>
          <interface>eth1</interface>
          <drop-unmatched>true</drop-unmatched>
        </dhcp-server>
      </relay>
    </dhcp>
  </vrf>
</config>

```

## DNS proxy

DNS proxy allows forwarding DNS queries.

Here is an example of DNS proxy configuration to forward DNS queries to the 192.168.0.254 server.

```

vrouters running config# vrf main
vrouters running vrf main# dns proxy
vrouters running proxy# forward server 192.168.0.254
vrouters running dns# commit

```

To display the DNS proxy state:

```
vrouters running config# show state vrf main dns proxy
proxy
  enabled true
  forward
    server 10.200.0.2
    ..
  ..
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouters running config# show config xml absolute vrf main dns proxy
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <dns xmlns="urn:6wind:vrouter/dns">
      <proxy>
        <enabled>true</enabled>
        <forward>
          <server>192.168.0.254</server>
        </forward>
      </proxy>
    </dns>
  </vrf>
</config>
```

To flush the proxy cache:

```
vrouters> cmd dns proxy clear-cache
```

#### See also:

The *DNS proxy command reference* and the *DNS proxy clear-cache description* for more details

### 3.1.13 Maximum Capacity Specifications

This section shows maximum sizes and numbers of various objects/features that can be configured on a Turbo Router.

**Note:** The limits and timings in this article are only *indicative* and may vary with hardware capacity. Here is the reference platform that was used to produce the following numbers:

|        |                                      |
|--------|--------------------------------------|
| CPU    | Intel® Xeon® Gold 6152 CPU @ 2.10GHz |
| Memory | 48GB                                 |

**Important:** The timings reported in the following table are given when configuring objects **all at once**: starting with 0 configured objects and creating N new ones in a single <edit-config> or commit operation.

Configuring objects **one by one** (or incrementally) will be *slower*. Processing the difference between the current configuration and the increment takes more time, the bigger the configuration is.

However, this limitation only becomes significant when reaching the upper limits. For reasonably small setups, you can assume that the configuration time is linear with the number of configured objects.

| Feature                        | Number of Objects | Configuration Time | Unconfiguration Time | Notes        |
|--------------------------------|-------------------|--------------------|----------------------|--------------|
| <i>Multiple logical VRF</i>    | 50                | 2 s                | 2 s                  | 1            |
| <i>Multiple logical VRF</i>    | 500               | 10 s               | 30 s                 | Page 2, 1    |
| <i>Bridge</i>                  | 500               | 10 s               | 40 s                 | 2, 3         |
| <i>Bridge</i>                  | 5000              | 60 s               | 600 s                | Page 3, 2, ? |
| <i>GRE</i>                     | 500               | 10 s               | 20 s                 | ?            |
| <i>GRE</i>                     | 5000              | 100 s              | 250 s                | ?            |
| <i>IPv4 and IPv6 tunneling</i> | 500               | 5 s                | 20 s                 | ?            |
| <i>IPv4 and IPv6 tunneling</i> | 5000              | 80 s               | 250 s                | ?            |
| <i>LAG</i>                     | 500               | 10 s               | 20 s                 | ?            |
| <i>LAG</i>                     | 5000              | 100 s              | 300 s                | ?            |
| <i>Loopback</i>                | 500               | 10 s               | 20 s                 |              |
| <i>Loopback</i>                | 5000              | 60 s               | 250 s                |              |
| <i>Static routes</i>           | 500               | 5 s                | 2 s                  |              |
| <i>Static routes</i>           | 5000              | 30 s               | 10 s                 |              |
| <i>SVTI</i>                    | 500               | 10 s               | 30 s                 | ?            |
| <i>SVTI</i>                    | 5000              | 60 s               | 250 s                | ?            |
| <i>veth, XVRF</i>              | 100               | 10 s               | 5 s                  | 4            |
| <i>veth, XVRF</i>              | 500               | 60 s               | 200 s                | ?            |
| <i>VLAN</i>                    | 500               | 15                 | 25 s                 | ?            |
| <i>VLAN</i>                    | 5000              | 150 s              | 250 s                | ?            |
| <i>VXLAN</i>                   | 500               | 15                 | 25 s                 | ?            |
| <i>VXLAN</i>                   | 5000              | 150 s              | 250 s                | ?            |

<sup>1</sup> VRFs are the most limiting feature. We do not recommend using more than 100 different ones on a single system.

<sup>2</sup> Bridges are much slower to delete than other interfaces.

<sup>3</sup> With all interfaces in the same VRF. Configuration will be slower when doing cross VRF tunnels.

<sup>4</sup> Cross VRF interfaces are inherently limited by the number of VRFs.

### 3.1.14 Troubleshooting

#### Troubleshooting Report

This command allows collecting various diagnostics of the system that can be provided to 6WIND support to help debugging a problem.

The troubleshooting report includes the following information:

- Linux networking information
  - ethtool on all known links
  - interfaces, addresses, routes, neighbours and IPsec
  - active network connections
  - Netfilter tables, bridge
- system information
  - topology
  - processors hierarchy
  - interrupts
  - memory
  - PCI peripherals
  - DMI/MBIOS
  - kernel version, logs, cmdline and loaded modules
  - distribution
  - services list
  - logs
  - processes list
  - cpuset
  - devices (/dev)
  - IRQ affinity
  - mounted partitions
- core dumps
- fast path information
  - configuration, version, logs, status
  - debug info (ports, tables, etc.)

- running and startup configurations
- license information
  - average network throughput (measured in Rx)
  - average and current number of IPsec tunnels
  - average and current number of CG-NAT connections

### List existing troubleshooting reports

```
vrouter> cmd troubleshooting-report list
NAME                               SIZE
2018-09-24_17-21-31.tgz           295.7K
2018-09-24_17-21-40.tgz           295.8K
vrouter>
```

### Create a new troubleshooting report

```
vrouter> cmd troubleshooting-report new
Gathering information. This may take some time...
Saved into /var/lib/yams/troubleshooting-reports/2018-09-24_17-27-07.tgz
vrouter>
```

### Delete an existing report

```
vrouter> cmd troubleshooting-report delete 2018-09-24_17-21-31.tgz
OK.
vrouter>
```

### Export an existing report to a remote location

```
vrouter> cmd troubleshooting-report export 2018-09-24_17-27-07.tgz url scp://
↪ john:s3cr3t@10.1.2.3/home/john
OK.
vrouter> cmd troubleshooting-report export 2018-09-24_17-27-07.tgz url smtp://10.
↪ 1.2.100/john@acme.com
OK.
vrouter>
```

It is possible to export via multiple protocols: FTP, HTTP, TFTP, SCP, SFTP and SMTP (for the later, you will need to specify an email address instead of a file path).

### Flush all existing reports

```
vrouter> cmd troubleshooting-report flush
OK.
vrouter>
```

**See also:**

The *command reference* for details.

**System****Operating system**

This context shows information about the machine operating system.

To display it:

```
vrouter> show state / system linux
linux
  cpu-usage cpu0
    busy 3
    ..
  cpu-usage cpu1
    busy 5
    ..
  cpu-usage cpu2
    busy 3
    ..
  cpu-usage cpu3
    busy 7
    ..
  memory
    available 4763774976
    total 5200089088
    ..
  disk-usage sda
    total 32212254720
    partition sda1
      label cloud_Boot
      fstype vfat
      total 104857600
      ..
    partition sda2
      label cloud_Releases
      fstype ext4
      total 9663676416
      available 19223638016
      ..
    partition sda3
      label cloud_Data
```

(continues on next page)

(continued from previous page)

```
fstype ext4
total 10737418240
available 9901813760
..
..
..
```

## Product

This context shows informations about the product.

To display the product name:

```
vrouter> show state / system product name
name "Turbo Router"
```

To display the product version:

```
vrouter> show state / system product version
version X.Y.Z
```

To display the license status:

```
vrouter> show state / system product license
license valid
```

## Log

Display system logs.

```
vrouter> show log [max-lines <NUM>]
```

To filter logs by service, facility, severity and/or VRF:

```
vrouter> show log [service <NAME>] [facility <NAME>] [level <LEVEL>] [vrf <NAME>]
```

---

**Note:** Each service has its own logging policy with regards to syslog facilities and severities. Refer to the services' documentation for details.

---

Example:

```

vrouters> show log service ntp vrf main max-lines 2
-- Logs begin at Tue 2018-09-25 18:23:28 CEST, end at Tue 2018-09-25 18:34:45 CEST. --
Sep 25 18:34:37 vrouter ntpd[1023]: Soliciting pool server 137.74.28.231
Sep 25 18:34:45 vrouter ntpd[1023]: Soliciting pool server 2001:67c:1560:8003::c7
vrouters> show log facility kernel level greater-or-equal warning
-- Logs begin at Tue 2018-09-25 18:23:28 CEST, end at Tue 2018-09-25 18:34:45 CEST. --
Sep 25 15:28:27 vrouter kernel: systemd-shutdown: 41 output lines suppressed due to
↳ratelimiting
-- Reboot --
Sep 25 11:51:58 vrouter kernel: #2
Sep 25 11:51:58 vrouter kernel: acpi PNP0A03:00: fail to add MMCONFIG information, can
↳'t access extended PCI configuration space under this bridge.
Sep 25 11:51:58 vrouter kernel: ACPI: PCI Interrupt Link [LNKC] enabled at IRQ 11

```

**See also:**

- The *command reference* for details about the API.
- The *Remote Log Filtering Configuration* for details about syslog facilities and severities.

**Identify A NIC Port**

If you ever need to, you can have a specific port of a physical NIC on the router blink to visually identify it.

To do that, run the following command:

```
vrouters> cmd identify-port pci-b131s0f1 duration 300
```

Where `pci-b131s0` is the `network-port` you want to identify.

The command can be interrupted before the specified duration (in seconds) by hitting `ctrl-c`.

---

**Note:** If you see the following error message:

```
Cannot identify NIC: Operation not supported
```

It means that your network adapter does not support LED control.

---

**Tip:** To display the list of all `network-ports` and their description, you may use the following command:

```
vrouters> show state / network-port
network-port pci-b6s0
pci-bus-addr 0000:06:00.0
vendor "Intel Corporation"
```

(continues on next page)

(continued from previous page)

```
model "I350 Gigabit Network Connection"
mac-address 52:54:00:12:34:57
interface eth0
..
network-port pci-b131s0
pci-bus-addr 0000:83:00.0
vendor "Intel Corporation"
model "82599ES 10-Gigabit SFI/SFP+ Network Connection"
mac-address de:ad:de:01:02:03
interface eth1
..
network-port pci-b131s0f1
pci-bus-addr 0000:83:00.1
vendor "Intel Corporation"
model "82599ES 10-Gigabit SFI/SFP+ Network Connection"
mac-address 52:54:00:12:34:56
interface eth2
..
network-port pci-b134s0
pci-bus-addr 0000:86:00.0
vendor "Mellanox Technologies"
model "MT27700 Family [ConnectX-4]"
mac-address 52:54:00:12:34:58
interface eth3
..
network-port pci-b134s0f1
pci-bus-addr 0000:86:00.1
vendor "Mellanox Technologies"
model "MT27700 Family [ConnectX-4]"
mac-address 52:54:00:12:34:59
interface eth4
..
```

**See also:**

The *command reference* for more details.

## Network

### Ping

To send ICMP ECHO\_REQUESTs to network hosts, you can use the following command:

```
vrouter> cmd ping host.domain.tld packetsize 256
PING host.domain.tld (10.0.2.2) 256(284) bytes of data.
264 bytes from host.domain.tld (10.0.2.2): icmp_seq=1 ttl=255 time=0.208 ms
264 bytes from host.domain.tld (10.0.2.2): icmp_seq=2 ttl=255 time=0.215 ms
264 bytes from host.domain.tld (10.0.2.2): icmp_seq=3 ttl=255 time=0.283 ms
264 bytes from host.domain.tld (10.0.2.2): icmp_seq=4 ttl=255 time=0.297 ms
^C
--- host.domain.tld ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.208/0.250/0.297/0.044 ms
vrouter>
```

The command can be interrupted by hitting `ctrl-c`.

#### See also:

The *command reference* for details.

## Traffic Capture

These commands enable displaying, capturing, managing and exporting network traffic flowing through a given network interface.

Display the network traffic flowing through a given network interface.

```
vrouter> cmd traffic-capture eth0 filter udp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
17:05:04.109799 de:ad:de:01:02:03 > 52:56:00:00:00:02, ethertype IPv6 (0x86dd), length 110: fec0::dcad:deff:fe01:203.123 > 2001:67c:1560:8003::c7.123: NTPv4, Client, length 48
17:05:11.109828 de:ad:de:01:02:03 > 52:55:0a:00:02:02, ethertype IPv4 (0x0800), length 90: 10.0.2.15.123 > 91.121.7.182.123: NTPv4, Client, length 48
17:05:13.109796 de:ad:de:01:02:03 > 52:56:00:00:00:02, ethertype IPv6 (0x86dd), length 110: fec0::dcad:deff:fe01:203.123 > 2001:bc8:2717:100::1.123: NTPv4, Client, length 48
^C
3 packets captured
3 packets received by filter
```

(continues on next page)

(continued from previous page)

```
0 packets dropped by kernel
```

```
vrouter>
```

The new argument enables capturing traffic in a given network interface:

```
vrouter> cmd traffic-capture new name traffic-eth0 filter udp eth0
tcpdump: listening on mgmt0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C3 packets captured
3 packets received by filter
0 packets dropped by kernel
vrouter>
```

Both commands can be interrupted by hitting ctrl-c.

#### See also:

- The *command reference* for details.
- The *command reference* for details.

It also is possible to list and flush all traffic captured saved in the machine or remove a specific one respectively with the `list`, `flush` and `delete` commands.

```
vrouter> cmd traffic-capture list
traffic-eth0
traffic-eth1
vrouter> cmd traffic-capture delete traffic-eth0
OK.
vrouter>
```

Then a specific capture can be read or exported with the `read` and `export` commands:

#### See also:

- The *command reference* for details about the `list` command.
- The *command reference* for details about the `delete` command.
- The *command reference* for details about the `flush` command.

```
vrouter> cmd traffic-capture read traffic-eth0
reading from file /var/lib/yams/traffic-captures/traffic-eth0.pcap, link-type EN10MB
↳(Ethernet)
17:05:04.109799 de:ad:de:01:02:03 > 52:56:00:00:00:02, ethertype IPv6 (0x86dd), length
↳110: fec0::dcad:deff:fe01:203.123 > 2001:67c:1560:8003::c7.123: NTPv4, Client,
↳length 48
17:05:11.109828 de:ad:de:01:02:03 > 52:55:0a:00:02:02, ethertype IPv4 (0x0800), length
↳90: 10.0.2.15.123 > 91.121.7.182.123: NTPv4, Client, length 48
```

(continues on next page)

(continued from previous page)

```

17:05:13.109796 de:ad:de:01:02:03 > 52:56:00:00:00:02, ethertype IPv6 (0x86dd), length 110
↳ 110: fec0::dcad:deff:fe01:203.123 > 2001:bc8:2717:100::1.123: NTPv4, Client, length 48
↳ 48
vrouters> cmd traffic-capture export traffic-eth0 url scp://user:passwd@host/tmp/
OK.
vrouters>

```

**See also:**

- The *command reference* for details about the read command.
- The *command reference* for details about the export command.

### 3.1.15 Automation

#### Cloud-init

Cloud-init handles early initialization of a cloud instance. More information is available at <https://cloudinit.readthedocs.io/en/latest/>.

Cloud-init is enabled by default. It can be disabled after the first boot using the following configuration. At the next reboot, cloud-init won't be called.

```

vrouters running # system cloud-init
vrouters running cloud-init# enabled false
vrouters running cloud-init# commit

```

To display cloud-init state:

```

vrouters running config# show state system cloud-init
cloud-init
  datasource "DataSourceNoCloud [seed=/dev/sr0][dsmode=local]"
  enabled true
  ..

```

The same configuration can be made using this NETCONF XML configuration:

```

vrouters running config# show config xml absolute system cloud-init
<config xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <cloud-init xmlns="urn:6wind:vrouter/system/cloud-init">
      <datasource>DataSourceNoCloud [seed=/dev/sr0][dsmode=local]</datasource>
      <enabled>true</enabled>
    </cloud-init>
  </system>
</config>

```

(continues on next page)

(continued from previous page)

```
</system>
</config>
```

**See also:**

The *command reference* for details.

**Remote configuration via NETCONF**

It is possible to remotely configure the equipment using the NETCONF protocol.

We will use `ncclient` as a NETCONF client. On another machine that will configure the router, install `ncclient` dependencies.

```
root@local# pip install xmldict ncclient
```

Create a `netconf.py` file with this content. This script will use the following NETCONF operations:

- lock, unlock
- get
- edit-config
- validate
- commit

It will configure the hostname twice, and it will check whether the system state has properly changed after each change.

```
#!/usr/bin/env python
# pip install xmldict ncclient

import json
from ncclient import manager
import time
import xmldict

def connect(host, user, password):
    conn = manager.connect(host=host,
                           username=user,
                           password=password,
                           timeout=10,
                           hostkey_verify=False)

    state = ""
```

(continues on next page)

(continued from previous page)

```

<nc:filter type="xpath"
  xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
  xmlns:vrouter="urn:6wind:vrouter"
  xmlns:vrouter-system="urn:6wind:vrouter/system"
  select="%s" />
""""

    conf = """
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <config xmlns="urn:6wind:vrouter">
    <system xmlns="urn:6wind:vrouter/system">
      <hostname>router</hostname>
    </system>
  </config>
</config>
""""

    new_hostname_conf = """
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <config xmlns="urn:6wind:vrouter">
    <system xmlns="urn:6wind:vrouter/system">
      <hostname>myhostname</hostname>
    </system>
  </config>
</config>
""""

    conn.lock()

    get_state = conn.get(state % '/vrouter:state/vrouter-system:system/hostname')
    print "***** hostname before configuration *****"
    print json.dumps(xmltodict.parse(get_state.data_xml), indent=2)

    print "***** set hostname to 'myhostname' *****"
    send_config = conn.edit_config(target='running', config=new_hostname_conf, default_
↵ operation='merge')

    check_config = conn.validate()

    conn.commit()

    get_state = conn.get(state % '/vrouter:state/vrouter-system:system/hostname')
    print "***** hostname is now 'myhostname' *****"
    print json.dumps(xmltodict.parse(get_state.data_xml), indent=2)

```

(continues on next page)

(continued from previous page)

```

print "***** revert to 'router' *****"
send_config = conn.edit_config(target='running', config=conf, default_operation=
↪ 'merge')

conn.commit()
get_state = conn.get(state % '/vrouter:state/vrouter-system:system/hostname')
print "***** hostname is now 'router' *****"
print json.dumps(xmltodict.parse(get_state.data_xml), indent=2)

conn.unlock()
conn.close_session()

if __name__ == '__main__':
    connect('<myip>', 'root', '<rootpass>')

```

Update the connect line to put the router IP, and the root password of the router, and launch the script. The following output is displayed.

```

# python netconf.py
***** hostname before configuration *****
{
  "data": {
    "@xmlns": "urn:ietf:params:xml:ns:netconf:base:1.0",
    "@xmlns:nc": "urn:ietf:params:xml:ns:netconf:base:1.0",
    "state": {
      "@xmlns": "urn:6wind:vrouter",
      "system": {
        "@xmlns": "urn:6wind:vrouter/system",
        "hostname": "router"
      }
    }
  }
}
***** set hostname to 'myhostname' *****
***** hostname is now 'myhostname' *****
{
  "data": {
    "@xmlns": "urn:ietf:params:xml:ns:netconf:base:1.0",
    "@xmlns:nc": "urn:ietf:params:xml:ns:netconf:base:1.0",
    "state": {
      "@xmlns": "urn:6wind:vrouter",
      "system": {
        "@xmlns": "urn:6wind:vrouter/system",

```

(continues on next page)

(continued from previous page)

```

        "hostname": "myhostname"
    }
}
}
}
***** revert to 'router' *****
***** hostname is now 'router' *****
{
  "data": {
    "@xmlns": "urn:ietf:params:xml:ns:netconf:base:1.0",
    "@xmlns:nc": "urn:ietf:params:xml:ns:netconf:base:1.0",
    "state": {
      "@xmlns": "urn:6wind:vrouter",
      "system": {
        "@xmlns": "urn:6wind:vrouter/system",
        "hostname": "router"
      }
    }
  }
}
}
}
}

```

## Ansible NETCONF Automation

Ansible supports configuring remote hosts using NETCONF (instead of the default SSH connection along with Linux shell commands). This guide explains how to leverage Ansible to configure multiple Turbo Router instances.

## Dependencies

This guide assumes that you have two (or more) Turbo Router instances that are booted and accessible on the network (NETCONF uses TCP port 830). Also, for clarity purposes, these machines should be reachable with their respective hostnames (thus, DNS or `/etc/hosts` must be configured accordingly).

To make sure it works, `ansible` version greater than 2.7.10 along with the `ncclient` and `jxmlease` python libraries are required. Here is how to install this in a python virtualenv:

```

$ python3 -m venv /tmp/ansible-netconf
$ . /tmp/ansible-netconf/bin/activate
$ which python
/tmp/ansible-netconf/bin/python
$ pip install -U pip setuptools wheel
...
Successfully installed pip-19.1.1 setuptools-41.0.1 wheel-0.33.4

```

(continues on next page)

(continued from previous page)

```
$ pip install "ansible > 2.7.10" ncclient jxmlease
...
Successfully installed MarkupSafe-1.1.1 PyYAML-5.1 ansible-2.8.0
asn1crypto-0.24.0 bcrypt-3.1.6 cffi-1.12.3 cryptography-2.6.1 jinja2-2.10.1
jxmlease-1.0.1 lxml-4.3.3 ncclient-0.6.4 paramiko-2.4.2 pyasn1-0.4.5
pyparser-2.19 pynacl-1.3.0 six-1.12.0
```

## Configuration

### Inventory

We need an “inventory” file that will reference all machines that we want to control with Ansible. Here we are using the YAML inventory format which is more readable than the default INI format.

```
# /tmp/ansible-netconf/hosts.yml
---
vrouters:
  vars:
    ansible_connection: netconf
    ansible_user: admin
    ansible_ssh_pass: admin      # using default admin user/password
    ansible_python_interpreter: python
  hosts:
    vrouter1:
      peer: vrouter2
      ifname: int0
      port: pci-b0s4
      ipaddr: 172.16.200.1
    vrouter2:
      peer: vrouter1
      ifname: ext0
      port: pci-b0s4
      ipaddr: 172.16.200.2
```

## Playbook

We also need to write a playbook. Here is a basic example that configures the hostname depending on the Ansible inventory name, and that configures a physical interface on both machines. Then, it runs the ping NETCONF RPC to check that the IP addresses have been properly configured on both machines.

```
# /tmp/ansible-netconf/playbook.yml
---
- hosts: vrouters
  gather_facts: false # facts gathering is not supported at the moment
  tasks:
    - name: fetch initial state
      netconf_get:
        display: json
        filter: "{{lookup('file', 'filter.xml')}}"
        register: state

    - name: print initial state
      debug:
        var: state.output.data

    - name: configure
      netconf_config:
        content: "{{lookup('template', 'config.xml')}}"

    - name: fetch state again
      netconf_get:
        display: json
        filter: "{{lookup('file', 'filter.xml')}}"
        register: state

    - name: print state after configuration has been applied
      debug:
        var: state.output.data

    - name: check connection both ways
      netconf_rpc:
        rpc: ping
        display: json
        xmlns: 'urn:6wind:vrouter/system'
        content: |
          <count>1</count>
          <destination>{{hostvars[peer].ipaddr}}</destination>
        register: ping
```

(continues on next page)

(continued from previous page)

```

- name: print ping outputs
debug:
  msg: "{{ping.output['nc:rpc-reply']['buffer'].splitlines()}}"

- name: unset hostname
netconf_config:
  content: |
    <config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <config xmlns="urn:6wind:vrouter">
        <system xmlns="urn:6wind:vrouter/system">
          <hostname xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
←nc:operation="delete"/>
          </system>
        </config>
      </config>

- name: change ipv4 address (not add a new one)
netconf_config:
  content: |
    <config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <config xmlns="urn:6wind:vrouter">
        <vrf>
          <name>main</name>
          <interface xmlns="urn:6wind:vrouter/interface">
            <physical>
              <name>{{ifname}}</name>
              <ipv4 xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
←nc:operation="replace">
                <address>
                  <ip>{{ipaddr}}/24</ip>
                </address>
              </ipv4>
            </physical>
          </interface>
        </vrf>
      </config>
    </config>

- name: fetch state again
netconf_get:
  display: json
  filter: "{{lookup('file', 'filter.xml')}}"
  register: state

```

(continues on next page)

(continued from previous page)

```

- name: print state after configuration has been modified
debug:
  var: state.output.data

- name: check connection both ways (again)
netconf_rpc:
  rpc: ping
  display: json
  xmlns: 'urn:6wind:vrouter/system'
  content: |
    <count>1</count>
    <destination>{{hostvars[peer].ipaddr}}00</destination>
register: ping

- name: print ping outputs
debug:
  msg: "{{ping.output['nc:rpc-reply']['buffer'].splitlines()}}"

```

**See also:**

The official Ansible documentation of the `netconf_get` ([https://docs.ansible.com/ansible/latest/modules/netconf\\_get\\_module.html](https://docs.ansible.com/ansible/latest/modules/netconf_get_module.html)), `netconf_config` ([https://docs.ansible.com/ansible/latest/modules/netconf\\_config\\_module.html](https://docs.ansible.com/ansible/latest/modules/netconf_config_module.html)) and `netconf_rpc` ([https://docs.ansible.com/ansible/latest/modules/netconf\\_rpc\\_module.html](https://docs.ansible.com/ansible/latest/modules/netconf_rpc_module.html)) modules.

Two additional XML files are referenced. They should be placed next to the playbook file itself.

**Config**

```

<!-- /tmp/ansible-netconf/config.xml -->
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <config xmlns="urn:6wind:vrouter">
    <system xmlns="urn:6wind:vrouter/system">
      <hostname>{{inventory_hostname}}</hostname>
    </system>
    <vrf>
      <name>main</name>
      <interface xmlns="urn:6wind:vrouter/interface">
        <physical>
          <name>{{ifname}}</name>
          <port>{{port}}</port>
          <ipv4>
            <address>
              <ip>{{ipaddr}}/24</ip>
            </address>
          </ipv4>
        </physical>
      </interface>
    </vrf>
  </config>
</config>

```

(continues on next page)

(continued from previous page)

```

    </address>
  </ipv4>
</physical>
</interface>
</vrf>
</config>
</config>

```

The structure of `config.xml` may be generated by running the following CLI commands:

```

localhost> edit running
localhost running config# system hostname vrouter2
localhost running config# vrf main interface physical ext0 port pci-b0s4 ipv4 address_
↪172.16.200.2/24
localhost running config# show config xml absolute nodefault
<config xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <hostname>vrouter2</hostname>
  </system>
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <physical>
        <name>ext0</name>
        <port>pci-b0s4</port>
        <ipv4>
          <address>
            <ip>172.16.200.2/24</ip>
          </address>
        </ipv4>
      </physical>
    </interface>
  </vrf>
</config>

```

---

**Important:** By default, the contents of the `<config>` XML node are *merged* with the current configuration. This is explained extensively in RFC 6241, Section 7.2. (<https://tools.ietf.org/html/rfc6241#section-7.2>).

In order to *replace* or *delete* some parts of the configuration, the operation XML attribute must be specified on the related XML nodes. The example playbook makes use of this attribute to unset a previously set hostname and replace an IPv4 address.

---

## Filter

```

<!-- /tmp/ansible-netconf/filter.xml -->
<state xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <hostname/>
    <product xmlns="urn:6wind:vrouter/system/product"/>
  </system>
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <physical>
        <name/>
        <ipv4>
          <address/>
        </ipv4>
        <port/>
        <oper-status/>
      </physical>
    </interface>
  </vrf>
</state>

```

The structure of `filter.xml` may be generated from combining the output of the following CLI commands:

```

localhost> show state xml absolute nodefault system
<state xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <hostname>localhost</hostname>
  ...
localhost> show state xml absolute nodefault vrf main interface physical ens3
<state xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <physical>
        <name>ens3</name>
        <ipv4>
          <address>
        ...

```

**Note:** The `playbook.yml` and `config.xml` files contain templating placeholders that will be replaced by respective host variables when the playbook is executed.

See Ansible official documentation ([https://docs.ansible.com/ansible/latest/user\\_guide/playbooks\\_templating.html](https://docs.ansible.com/ansible/latest/user_guide/playbooks_templating.html))

for more details.

---

## Execution

Once all these files are created, you may run `ansible-playbook` as follows:

```
$ ansible-playbook -i /tmp/ansible-netconf/hosts.yml /tmp/ansible-netconf/playbook.yml

PLAY [vrouters] *****

TASK [fetch initial state] *****
ok: [vrouter1]
ok: [vrouter2]

TASK [print initial state] *****
ok: [vrouter2] => {
  "state.output.data": {
    "state": {
      "system": {
        "hostname": "localhost",
        "product": {
          "license": "valid",
          "name": "Turbo Router",
          "version": "3.2"
        }
      },
      "vrf": {
        "interface": {
          "physical": [
            {
              "ipv4": {
                "address": {
                  "ip": "10.0.2.15/24"
                }
              },
              "name": "ens3",
              "oper-status": "UP",
              "port": "pci-b0s3"
            },
            {
              "name": "ens4",
              "oper-status": "DOWN",
              "port": "pci-b0s4"
            }
          ]
        }
      }
    }
  }
}
```

(continues on next page)



(continued from previous page)

```
TASK [configure] *****
changed: [vrouter2]
changed: [vrouter1]

TASK [fetch state again] *****
ok: [vrouter1]
ok: [vrouter2]

TASK [print state after configuration has been applied] *****
ok: [vrouter2] => {
  "state.output.data": {
    "state": {
      "system": {
        "hostname": "vrouter2",
        "product": {
          "license": "valid",
          "name": "Turbo Router",
          "version": "3.2"
        }
      },
      "vrf": {
        "interface": {
          "physical": [
            {
              "ipv4": {
                "address": {
                  "ip": "10.0.2.15/24"
                }
              },
              "name": "ens3",
              "oper-status": "UP",
              "port": "pci-b0s3"
            },
            {
              "ipv4": {
                "address": {
                  "ip": "172.16.200.2/24"
                }
              },
              "name": "ext0",
              "oper-status": "UP",
              "port": "pci-b0s4"
            }
          ]
        }
      }
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

        ]
      },
      "name": "main"
    }
  }
}
ok: [vrouter1] => {
  "state.output.data": {
    "state": {
      "system": {
        "hostname": "vrouter1",
        "product": {
          "license": "valid",
          "name": "Turbo Router",
          "version": "3.2"
        }
      },
      "vrf": {
        "interface": {
          "physical": [
            {
              "ipv4": {
                "address": {
                  "ip": "10.0.2.15/24"
                }
              },
              "name": "ens3",
              "oper-status": "UP",
              "port": "pci-b0s3"
            },
            {
              "ipv4": {
                "address": {
                  "ip": "172.16.200.1/24"
                }
              },
              "name": "int0",
              "oper-status": "UP",
              "port": "pci-b0s4"
            }
          ]
        }
      }
    }
  }
}

```

(continues on next page)

(continued from previous page)

```
    }
  }
}

TASK [check connection both ways] *****
ok: [vrouter1]
ok: [vrouter2]

TASK [print ping outputs] *****
ok: [vrouter2] => {
  "msg": [
    "PING 172.16.200.1 (172.16.200.1) 56(84) bytes of data.",
    "64 bytes from 172.16.200.1: icmp_seq=1 ttl=64 time=0.652 ms",
    "",
    "--- 172.16.200.1 ping statistics ---",
    "1 packets transmitted, 1 received, 0% packet loss, time 0ms",
    "rtt min/avg/max/mdev = 0.652/0.652/0.652/0.000 ms"
  ]
}
ok: [vrouter1] => {
  "msg": [
    "PING 172.16.200.2 (172.16.200.2) 56(84) bytes of data.",
    "64 bytes from 172.16.200.2: icmp_seq=1 ttl=64 time=0.758 ms",
    "",
    "--- 172.16.200.2 ping statistics ---",
    "1 packets transmitted, 1 received, 0% packet loss, time 0ms",
    "rtt min/avg/max/mdev = 0.758/0.758/0.758/0.000 ms"
  ]
}

TASK [unset hostname] *****
changed: [vrouter2]
changed: [vrouter1]

TASK [change ipv4 address (not add a new one)] *****
changed: [vrouter2]
changed: [vrouter1]

TASK [fetch state again] *****
ok: [vrouter1]
ok: [vrouter2]

TASK [print state after configuration has been modified] *****
```

(continues on next page)

(continued from previous page)

```
ok: [vrouter1] => {
  "state.output.data": {
    "state": {
      "system": {
        "hostname": "vrouter1",
        "product": {
          "license": "unknown",
          "name": "Turbo Router",
          "version": "3.2"
        }
      },
      "vrf": {
        "interface": {
          "physical": [
            {
              "ipv4": {
                "address": {
                  "ip": "10.0.2.15/24"
                }
              },
              "name": "ens3",
              "oper-status": "UP",
              "port": "pci-b0s3"
            },
            {
              "ipv4": {
                "address": {
                  "ip": "172.16.200.100/24"
                }
              },
              "name": "int0",
              "oper-status": "UP",
              "port": "pci-b0s4"
            }
          ]
        },
        "name": "main"
      }
    }
  }
}
ok: [vrouter2] => {
  "state.output.data": {
    "state": {
```

(continues on next page)

(continued from previous page)

```
"system": {
  "hostname": "vrouter2",
  "product": {
    "license": "unknown",
    "name": "Turbo Router",
    "version": "3.2"
  }
},
"vrf": {
  "interface": [
    {
      "physical": [
        {
          "ipv4": {
            "address": {
              "ip": "10.0.2.15/24"
            }
          },
          "name": "ens3",
          "oper-status": "UP",
          "port": "pci-b0s3"
        },
        {
          "ipv4": {
            "address": {
              "ip": "172.16.200.200/24"
            }
          },
          "name": "ext0",
          "oper-status": "UP",
          "port": "pci-b0s4"
        }
      ]
    }
  ],
  "name": "main"
}
}

TASK [check connection both ways (again)] *****
ok: [vrouter1]
ok: [vrouter2]

TASK [print ping outputs] *****
```

(continues on next page)

(continued from previous page)

```

ok: [vrouter1] => {
  "msg": [
    "PING 172.16.200.200 (172.16.200.200) 56(84) bytes of data.",
    "64 bytes from 172.16.200.200: icmp_seq=1 ttl=64 time=1.07 ms",
    "",
    "--- 172.16.200.200 ping statistics ---",
    "1 packets transmitted, 1 received, 0% packet loss, time 0ms",
    "rtt min/avg/max/mdev = 1.076/1.076/1.076/0.000 ms"
  ]
}
ok: [vrouter2] => {
  "msg": [
    "PING 172.16.200.100 (172.16.200.100) 56(84) bytes of data.",
    "64 bytes from 172.16.200.100: icmp_seq=1 ttl=64 time=10.1 ms",
    "",
    "--- 172.16.200.100 ping statistics ---",
    "1 packets transmitted, 1 received, 0% packet loss, time 0ms",
    "rtt min/avg/max/mdev = 10.119/10.119/10.119/0.000 ms"
  ]
}

PLAY RECAP *****
vrouter1: ok=13  changed=3  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
vrouter2: ok=13  changed=3  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

```

## Additional examples

To copy the running configuration in startup, use:

```

- name: copy running startup
  netconf_rpc:
    rpc: copy-config
    xmlns: 'urn:ietf:params:xml:ns:netconf:base:1.0'
    content: |
      <target><startup/></target>
      <source><running/></source>

```

To delete the startup configuration, use:

```

- name: delete startup
  netconf_rpc:
    rpc: delete-config
    xmlns: 'urn:ietf:params:xml:ns:netconf:base:1.0'

```

(continues on next page)

(continued from previous page)

```
content: |  
  <target><startup/></target>
```

## 3.2 Command Reference

### 3.2.1 cmd

Execute remote commands on the NETCONF server.

#### cmd banner

```
vrouter> cmd banner pre-login [message <string>] [reset]  
vrouter> cmd banner post-login [message <string>] [reset]
```

Manage login banner.

#### Input Parameters

**pre-login [message <string>] [reset]** Manage banner before a user logs in.

**message <string>** Message to display.

**reset** Reset message to factory defaults.

**post-login [message <string>] [reset]** Manage banner after a user logs in.

**message <string>** Message to display.

**reset** Reset message to factory defaults.

#### cmd reboot

```
vrouter> cmd reboot [delay <uint32>] [cancel] [force]
```

Schedule a system reboot after a grace period.

## Input Parameters

**delay** <uint32> The number of seconds to wait before reboot. During that time, it is possible to cancel the reboot.

**cancel** If defined, cancel a pending reboot.

**force** If defined, force reboot even if startup configuration is different than running configuration.

### cmd poweroff

```
vrouter> cmd poweroff [delay <uint32>] [cancel] [force]
```

Schedule a system poweroff after a grace period.

## Input Parameters

**delay** <uint32> The number of seconds to wait before poweroff. During that time, it is possible to cancel the poweroff.

**cancel** If defined, cancel a pending poweroff.

**force** If defined, force poweroff even if startup configuration is different than running configuration.

### cmd ping

```
vrouter> cmd ping [vrf <string>] [count <uint16>] [packetsize <uint16>] [nodns] \
... [ipv6] [source <string>] [rate <uint16>] <destination>
```

Send ICMP ECHO\_REQUEST messages to network hosts and print their responses.

## Input Parameters

**vrf** <string> The VRF in which to send the ICMP ECHO\_REQUESTs. By default, they are sent in the 'main' vrf.

**count** <uint16> Stop after sending count ECHO\_REQUEST packets.

**packetsize** <uint16> Specifies the number of data bytes to be sent. The default is 56, which translates into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.

**nodns** Numeric output only. No attempt will be made to lookup symbolic names for host addresses.

**ipv6** Force IPv6 operation only. By default, it is detected from the destination. If destination is a host name, ipv4 is used by default unless this flag is set.

**source <string>** Either an address, or an interface name. If interface is an address, it sets source address to specified interface address. If interface in an interface name, it sets source interface to specified interface. For IPv6, when doing ping to a link-local scope address, link specification (by the ‘%’-notation in destination, or by this option) is required.

**rate <uint16>** The number of packets to send per second. By default, 1 packet is sent every second.

**<destination> (mandatory)** The destination host (name or IP address).

### cmd traceroute

```
vrouter> cmd traceroute [vrf <string>] [nodns] [ipv6] [source SOURCE] [source-
↪interface <string>] \
...          <host>
```

Display the route (path) that was used to connect to a certain IP address or hostname. It also measures the transit delays among hops.

### Input Parameters

**vrf <string>** The VRF in which the packets are sent by traceroute. By default, they are sent in the ‘main’ vrf.

**nodns** Do not try to map IP addresses to host names when displaying them.

**ipv6** Force IPv6 operation only. By default, it is detected from the destination. If destination is a host name, ipv4 is used by default unless this flag is set.

**source SOURCE** Chooses an alternative source address. Note that an address of one of the interfaces must be selected. By default, the address of the outgoing interface is used.

| SOURCE values | Description      |
|---------------|------------------|
| <A.B.C.D>     | An IPv4 address. |
| <X:X::X:X>    | An IPv6 address. |

**source-interface <string>** Specifies the interface through which traceroute should send packets. By default, the interface is selected according to the routing table.

**<host> (mandatory)** The destination host (name or IP address).

## cmd show-traffic

```
vrouter> cmd show-traffic [vrf <string>] [count <uint16>] [filter <pcap-expr>] <ifname>
```

Print traffic flowing on a network interface.

### Input Parameters

**vrf <string>** The VRF in which to capture traffic. This must be the VRF the interface belongs to. By default, the interface is assumed to be in the ‘main’ vrf.

**count <uint16>** Stop after capturing count packets.

**filter <pcap-expr>** Optional filter expression. This must be a valid PCAP filter. See <https://www.tcpdump.org/manpages/pcap-filter.7.html> for more details.

**<ifname>** (mandatory) The name of the network interface on which to monitor traffic.

## cmd traffic-capture

```
vrouter> cmd traffic-capture [vrf <string>] [count <uint16>] [filter <pcap-expr>]  
↪<ifname>
```

Print traffic flowing on a network interface.

### Input Parameters

**vrf <string>** The VRF in which to capture traffic. This must be the VRF the interface belongs to. By default, the interface is assumed to be in the ‘main’ vrf.

**count <uint16>** Stop after capturing count packets.

**filter <pcap-expr>** Optional filter expression. This must be a valid PCAP filter. See <https://www.tcpdump.org/manpages/pcap-filter.7.html> for more details.

**<ifname>** (mandatory) The name of the network interface on which to monitor traffic.

### cmd traffic-capture new

```
vrouter> cmd traffic-capture new [name <name>] [vrf <string>] [count <uint16>] [filter  
↳<pcap-expr>] \  
...      <ifname>
```

Capture traffic flowing on a network interface.

### Input Parameters

**name <name>** The name of the capture file. If not set a unique name will be automatically chosen (in format YYYY-MM-DD\_HH-MM-SS.<ifname>.pcap). otherwise, if the file already exists it will be overwritten.

**vrf <string>** The VRF in which to capture traffic. This must be the VRF the interface belongs to. By default, the interface is assumed to be in the 'main' vrf.

**count <uint16>** Stop after capturing count packets.

**filter <pcap-expr>** Optional filter expression. This must be a valid PCAP filter. See <https://www.tcpdump.org/manpages/pcap-filter.7.html> for more details.

**<ifname> (mandatory)** The name of the network interface on which to monitor traffic.

### cmd traffic-capture list

```
vrouter> cmd traffic-capture list
```

List captured traffic flow.

### cmd traffic-capture read

```
vrouter> cmd traffic-capture read <name>
```

Read a captured traffic flow.

### Input Parameters

**<name> (mandatory)** The name of the capture to read.

## cmd traffic-capture export

```
vrouter> cmd traffic-capture export [vrf <string>] url URL [user <string>] [password
↳<string>] <name>
```

Export a captured traffic flow.

### Input Parameters

**vrf <string>** The VRF in which remote access is done. By default, they are sent in the ‘main’ vrf.

**url URL (mandatory)** The destination URL.

| URL values  | Description   |
|---|---|
| <sftp://[user[:password]@host[:port]/path/to/file>    | An SFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.    |
| <scp://[user[:password]@host[:port]/path/to/file>     | A SCP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.      |
| <smtp[s]://[user[:password]@host[:port]/path/to/file> | An SMTPS email URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.  |
| <ftp://[user[:password]@host[:port]/path/to/file>     | An FTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.     |
| <tftp://host[:port]/path/to/file>                     | A TFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2].   |
| <http[s]://[user[:password]@host[:port]/path/to/file> | An HTTP(S) file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields. |

**user <string>** The URL user name (not percent-encoded). If specified, the user name should not be included in the URL.

**password <string>** The URL password (not percent-encoded). If specified, the user name should not be included in the URL.

**<name>** (mandatory) The name of the capture to export.

### cmd traffic-capture flush

```
vrouter> cmd traffic-capture flush
```

Flush all captured traffic flow.

### cmd traffic-capture delete

```
vrouter> cmd traffic-capture delete <name>
```

Delete a captured traffic flow.

### Input Parameters

**<name>** (mandatory) The name of the capture to delete.

### cmd identify-port

```
vrouter> cmd identify-port NAME [duration <uint16>]
```

Initiate adapter-specific action intended to enable an operator to easily identify a physical network interface by sight. Typically this involves blinking one or more LEDs on the specific network port.

### Input Parameters

**NAME** (mandatory) The port name.

|      |                |
|------|----------------|
| NAME | PCI port name. |
|------|----------------|

**duration** <uint16> Length of time to perform the identification, in seconds.

## cmd system-image

```
vrouter> cmd system-image install-on-disk [backup-url BACKUP-URL] [user <string>] \
↳[password <string>] \
...      <device>
vrouter> cmd system-image import [name <name>] [vrf <string>] [user <string>] \
↳[password <string>] \
...      URL
vrouter> cmd system-image delete <name>
vrouter> cmd system-image list
vrouter> cmd system-image rename <name> new-name <string>
vrouter> cmd system-image set-default [<name>]
vrouter> cmd system-image set-next [<name>]
```

Manage system images.

### Input Parameters

**install-on-disk** [backup-url BACKUP-URL] [user <string>] [password <string>] <device>

Install the system on a specific device.

**backup-url** BACKUP-URL The URL where the backup files are stored.

| BACKUP-URL values                                     | Description   |
|---|---|
| <sftp://[user[:password]@host[:port]/path/to/file>    | An SFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.    |
| <scp://[user[:password]@host[:port]/path/to/file>     | An SCP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.     |
| <ftp://[user[:password]@host[:port]/path/to/file>     | An FTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.     |
| <tftp://host[:port]/path/to/file>                     | A TFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2].   |
| <http[s]://[user[:password]@host[:port]/path/to/file> | An HTTP(S) file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields. |

**user** <string> The URL user name (not percent-encoded). If specified, the user name should not be included in the URL.

**password** <string> The URL password (not percent-encoded). If specified, the user name should not be included in the URL.

**<device>** (mandatory) The device on which to install the currently booted image.

**import** [name <name>] [vrf <string>] [user <string>] [password <string>] **URL** Import a new system .update image from a remote URL.

**name** <name> The custom name to assign of the .update image.

**vrf** <string> The VRF in which remote access is done. By default, they are sent in the 'main' vrf.

**user** <string> The URL user name (not percent-encoded). If specified, the user name should not be included in the URL.

**password** <string> The URL password (not percent-encoded). If specified, the user name should not be included in the URL.

**URL** (mandatory) The URL from which to download the .update image.

| URL values   | Description  |
|--|--|
| <http[s]://[user[:password]@]host[:port]/path/to/file.update>  | An HTTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields. |
| <sftp://[user[:password]@]host[:port]/path/to/file.update>     | An SFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields. |
| <scp://[user[:password]@]host[:port]/path/to/file.update>      | An SCP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.  |
| <ftp://{[user{[:password]}]@host{[:port]}/path/to/file.update> | An FTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.  |
| <tftp://host{[:port]}/path/to/file.update>                     | A TFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2].  |

**delete** <name> Delete an imported image.

**<name>** (mandatory) The name of the image to delete.

**list** Display a list of imported images.

**rename** <name> new-name <string> Rename an image.

**<name>** (mandatory) The current name of the image.

**new-name** <string> (mandatory) The new name of the image.

**set-default** [<name>] Set a system image as default boot image.

**<name>** The name of the image to set as default.

**set-next** [<name>] Set a system image as next boot image. If it does not boot, the system will reboot to the default image. This option is not available for LVM disks.

**<name>** The name of the image to set as next.

### cmd backup

```
vrouters> cmd backup import [vrf <string>] url URL [user <string>] [password <string>]
vrouters> cmd backup export [vrf <string>] url URL [user <string>] [password <string>]
```

Import/export backup archives containing configurations, keys, certificates, licenses.

### Input Parameters

**import [vrf <string>] url URL [user <string>] [password <string>]** Import backup archive from a remote server. WARNING: it will overwrite current configurations.

**vrf <string>** The VRF in which remote access is done. By default, they are sent in the ‘main’ vrf.

**url URL (mandatory)** The source URL.

| URL values  | Description   |
|---|---|
| <sftp://[user[:password]@host[:port]/path/to/file>    | An SFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: ‘?’ becomes ‘%3f’). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.    |
| <scp://[user[:password]@host[:port]/path/to/file>     | An SCP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: ‘?’ becomes ‘%3f’). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.     |
| <ftp://[user[:password]@host[:port]/path/to/file>     | An FTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: ‘?’ becomes ‘%3f’). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.     |
| <tftp://host[:port]/path/to/file>                     | A TFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2].   |
| <http[s]://[user[:password]@host[:port]/path/to/file> | An HTTP(S) file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: ‘?’ becomes ‘%3f’). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields. |

**user <string>** The URL user name (not percent-encoded). If specified, the user name should not be included in the URL.

**password <string>** The URL password (not percent-encoded). If specified, the user name should not be included in the URL.

**export [vrf <string>] url URL [user <string>] [password <string>]** Export backup archive to a remote server.

**vrf <string>** The VRF in which remote access is done. By default, they are sent in the ‘main’ vrf.

**url URL (mandatory)** The destination URL.

| URL values  | Description   |
|---|---|
| <sftp://[user[:password]@host[:port]/path/to/file>    | An SFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: ‘?’ becomes ‘%3f’). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.    |
| <scp://[user[:password]@host[:port]/path/to/file>     | A SCP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: ‘?’ becomes ‘%3f’). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.      |
| <smtp[s]://[user[:password]@host[:port]]>             | An SMTP email URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: ‘?’ becomes ‘%3f’). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.   |
| <ftp://[user[:password]@host[:port]/path/to/file>     | An FTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: ‘?’ becomes ‘%3f’). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.     |
| <tftp://host[:port]/path/to/file>                     | A TFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2].   |
| <http[s]://[user[:password]@host[:port]/path/to/file> | An HTTP(S) file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: ‘?’ becomes ‘%3f’). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields. |

**user <string>** The URL user name (not percent-encoded). If specified, the user name should not be included in the URL.

**password <string>** The URL password (not percent-encoded). If specified, the user name should not be included in the URL.

## cmd set-next-boot-params

```
vrouter> cmd set-next-boot-params [intel-iommu true|false] [iommu-allow-unsafe-
↳interrupts true|false] \
...          [ixgbe-allow-unsupported-sfp true|false] [isolate-cpus ISOLATE-CPUS]
```

Set boot parameters, taking effect at next reboot. Image must be installed on disk.

## Input Parameters

**intel-iommu true|false** Enable intel iommu driver. Control intel\_iommu=on|off kernel option.

**iommu-allow-unsafe-interrupts true|false** Enable PCI passthrough on hardware that does not support interrupt remapping, when VM are trusted. Control vfio\_iommu\_type1.allow\_unsafe\_interrupts=0|1 kernel option.

**ixgbe-allow-unsupported-sfp true|false** Bypass SFPs types restrictions on Intel ixgbe NICs. Control ixgbe.allow\_unsupported\_sfp=0|1 kernel option.

**isolate-cpus ISOLATE-CPUS** Isolate cpus from kernel threads, rcu callbacks, and reduce the scheduler ticks. A good value for this parameter is the fast path coremask.

| ISOLATE-CPUS values | Description   |
|---------------------|---|
| <cores-list>        | A comma-separated list of cores or core ranges. Example: '1,4-7,10-12'. |
| none                | Unset the coremask.   |

## cmd license certificate

```
vrouter> cmd license certificate import [url URL] [user <string>] [password <string>] \
↳[content <string>] \
...          serial <string>
vrouter> cmd license certificate list
vrouter> cmd license certificate delete <string>
vrouter> cmd license certificate request-activation serial <string>
vrouter> cmd license certificate request-deactivation serial <string>
vrouter> cmd license certificate cancel-request
```

Manage license certificate requests for offline activation through an activation webpage.

## Input Parameters

**import [url URL] [user <string>] [password <string>] [content <string>] serial <string>**  
 Import a license certificate. It will be used to activate the license on the device. The certificate will be deleted when the first configuration using it is committed.

**url URL** The URL from which to download the license certificate.

| URL values  | Description   |
|---|---|
| <http[s]://[user[:password@]host[:port]/path/to/file> | An HTTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!@\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields. |
| <sftp://[user[:password@]host[:port]/path/to/file>    | An SFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!@\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields. |
| <scp://[user[:password@]host[:port]/path/to/file>     | An SCP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!@\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.  |
| <ftp://[user[:password@]host[:port]/path/to/file>     | An FTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!@\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.  |
| <tftp://host[:port]/path/to/file>                     | A TFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2].   |

**user <string>** The URL user name (not percent-encoded). If specified, the user name should not be included in the URL.

**password <string>** The URL password (not percent-encoded). If specified, the user name should not be included in the URL.

**content <string>** The raw contents of the license certificate.

**serial <string> (mandatory)** The serial number associated with the license certificate.

**list** List downloaded license certificates that not are consumed yet.

**delete <string>** Delete a license certificate.

**<string> (mandatory)** The name of license certificate to delete.

**request-activation serial <string>** Request an activation certificate for a serial number. The resulting certificate must then be entered on the Licensing User Portal, in the Offline Activation tab. The resulting

certificate must be imported using the `cmd license certificate import` command. The licensing has to be disabled in configuration to use this rpc.

**serial <string> (mandatory)** The serial number associated to this activation request.

**request-deactivation serial <string>** Request an deactivation certificate for a serial number. The resulting certificate must then be entered on the Licensing User Portal, in the Offline Activation tab. Requesting the certificate will disable the license on this device. The licensing has to be disabled in configuration to use this rpc.

**serial <string> (mandatory)** The serial number associated to this deactivation request.

**cancel-request** Cancel any request in progress.

### cmd license file

```
vrouter> cmd license file import [url URL] [user <string>] [password <string>]
↳[content <string>] \
...          serial <string>
vrouter> cmd license file list
vrouter> cmd license file delete <string>
```

Manage license files.

### Input Parameters

**import [url URL] [user <string>] [password <string>] [content <string>] serial <string>**  
Import a license file.

**url URL** The URL from which to download the license file.

| URL values  | Description  |
|---|--|
| <http[s]://[user[:password]@host[:port]/path/]file> | An HTTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields. |
| <sftp://[user[:password]@host[:port]/path/]file>    | An SFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields. |
| <scp://[user[:password]@host[:port]/path/]file>     | An SCP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.  |
| <ftp://[user[:password]@host[:port]/path/]file>     | An FTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.  |
| <tftp://host[:port]/path/to/file>                   | A TFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2].  |

**user <string>** The URL user name (not percent-encoded). If specified, the user name should not be included in the URL.

**password <string>** The URL password (not percent-encoded). If specified, the user name should not be included in the URL.

**content <string>** The raw contents of the license file.

**serial <string> (mandatory)** The serial number associated with the license file. It will be used as reference in the configuration.

**list** List downloaded license files.

**delete <string>** Delete a license file.

**<string> (mandatory)** The name of license file to delete.

## cmd license refresh

```
vrouter> cmd license refresh
```

Refresh the license.

## cmd troubleshooting-report

```
vrouter> cmd troubleshooting-report list
vrouter> cmd troubleshooting-report delete <name>
vrouter> cmd troubleshooting-report flush
vrouter> cmd troubleshooting-report new
vrouter> cmd troubleshooting-report export [vrf <string>] url URL [user <string>] \
↪[password <string>] \
...          <name>
```

Manage troubleshooting reports.

## Input Parameters

**list** List existing troubleshooting reports.

**delete <name>** Delete an existing troubleshooting report.

**<name> (mandatory)** The name of the report to delete.

**flush** Delete all existing troubleshooting reports.

**new** Generate a new troubleshooting report.

**export [vrf <string>] url URL [user <string>] [password <string>] <name>** Export an existing troubleshooting report to a remote server via SFTP.

**vrf <string>** The VRF in which remote access is done. By default, they are sent in the 'main' vrf.

**url URL (mandatory)** The destination URL.

| URL values   | Description   |
|--|---|
| <sftp://[user[:password]@host[:port]/path/IPv6]>                 | An SFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.      |
| <scp://[user[:password]@host[:port]/path/IPv6]>                  | An SCP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.       |
| <smtp[s]://[user[:password]@host[:port]/path/IPv6]>              | An SMTP (S) email URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields. |
| <ftp://[{}user[{}:password[{}]@{}host[{}:port[{}]/path/to/file]> | An FTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.       |
| <tftp://host[{}:port[{}]/path/to/file>                           | A TFTP file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2].   |
| <http[s]://[user[:password]@host[:port]/path/IPv6]>              | An HTTP(S) file URL. IPv6 addresses must be surrounded by square brackets [1234:bada::2]. The :/?#[!\$&'()*+;= characters in the user and password must be percent-encoded (e.g: '?' becomes '%3f'). See RFC 3986 section 2.1. For convenience, you should use the separate user and password fields.   |

**user <string>** The URL user name (not percent-encoded). If specified, the user name should not be included in the URL.

**password <string>** The URL password (not percent-encoded). If specified, the user name should not be included in the URL.

**<name> (mandatory)** The name of the report to export.

### cmd dns proxy clear-cache

```
vrouters> cmd dns proxy clear-cache [vrf <string>]
```

Clear DNS proxy cache.

## Input Parameters

**vrf** <string> Specify the VRF.

### cmd dhcp-client renew-lease

```
vrouter> cmd dhcp-client renew-lease [vrf <string>] IFNAME
```

Renew DHCP client lease period.

## Input Parameters

**vrf** <string> Specify the VRF.

**IFNAME** (mandatory) The interface name.

|        |                    |
|--------|--------------------|
| IFNAME | An interface name. |
|--------|--------------------|

### cmd bgp rpki ssh-key create

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> cmd bgp rpki ssh-key create type TYPE name <string>
```

Create SSH keys.

## Input Parameters

**type** **TYPE** (mandatory) SSH key type.

| TYPE values | Description          |
|-------------|----------------------|
| rsa-1024    | RSA in 1024 bits.    |
| rsa-2048    | RSA in 2048 bits.    |
| rsa-4096    | RSA in 4096 bits.    |
| ecdsa-256   | ECDSA in 256 bits.   |
| ecdsa-384   | ECDSA in 384 bits.   |
| ecdsa-521   | ECDSA in 521 bits.   |
| ed25519     | EDDSA in 25519 bits. |

**name** <string> (mandatory) Name of the new key pair.

## cmd bgp rpki ssh-key list

---

**Note:** requires a Turbo Router Network License.

---

```
vrouters> cmd bgp rpki ssh-key list [detail]
```

List SSH keys.

### Input Parameters

**detail** Show public key.

## cmd bgp rpki ssh-key delete

---

**Note:** requires a Turbo Router Network License.

---

```
vrouters> cmd bgp rpki ssh-key delete <string>
```

Delete SSH keys.

### Input Parameters

**<string>** (mandatory) Delete an existing key pair.

## cmd bgp rpki ssh-host add

---

**Note:** requires a Turbo Router Network License.

---

```
vrouters> cmd bgp rpki ssh-host add HOST [port PORT] [vrf VRF]
```

Add host to routing known hosts.

## Input Parameters

**HOST (mandatory)** Host name to add to known hosts.

|             |   |
|-------------|---|
| <b>HOST</b> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names <b>MUST</b> be encoded in punycode as described in RFC 3492. |
|-------------|---|

**port PORT** Use a specific port to join the remote host.

|             |   |
|-------------|---|
| <b>PORT</b> | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------------|---|

**vrf VRF** Specify the VRF.

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

### cmd **bgp rpki ssh-host delete**

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> cmd bgp rpki ssh-host delete HOST-NAME
```

Delete host from routing known hosts.

## Input Parameters

**HOST-NAME (mandatory)** Host name to remove from known hosts.

|           |   |
|-----------|---|
| HOST-NAME | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names <b>MUST</b> be encoded in punycode as described in RFC 3492. |
|-----------|---|

### 3.2.2 show

#### show summary

```
vrouter> show summary
```

Show a summary of the system state.

#### show interface

```
vrouter> show interface [vrf <string>] [type <identityref>] [LEVEL] [name <string>]
```

Show interface information.

## Input Parameters

**vrf <string>** VRF to look into.

**type <identityref>** Interface type.

**LEVEL** The level of information requested.

| LEVEL values                | Description   |
|-----------------------------|---|
| statistics                  | Display statistics.   |
| details                     | Display more details.                                       |
| up                          | Display UP interfaces only.                                 |
| hardware-statistics         | Display hardware statistics. Implies physical type.         |
| hardware-features           | Display hardware features. Implies physical type.           |
| hardware-information        | Display hardware information. Implies physical type.        |
| hardware-driver-information | Display hardware driver information. Implies physical type. |

**name** <string> Display only one interface by this name.

### show interface throughput

```
vrouters> show interface throughput [vrf <string>] [type <identityref>] [name <string>]
↳[count <uint16>] \
... [raw]
```

Show interface throughput every second.

### Input Parameters

**vrf** <string> VRF to look into.

**type** <identityref> Select all interfaces of this type.

**name** <string> Select this specific interface (may be specified multiple times).

**count** <uint16> Stop after the given number of seconds. By default, the throughput is displayed every second until the command is interrupted with `ctrl-c`.

**raw** Show the exact number of packets/bits received/transmitted every second instead of human readable values.

### show ipv4-routes

```
vrouters> show ipv4-routes [vrf <string>] [protocol <identityref>] [table <uint32>] [to_
↳T0]
```

Show IPv4 routing table.

## Input Parameters

**vrf** <string> Specify the VRF.

**protocol** <identityref> Filter routes by protocol.

**table** <uint32> Non-main Kernel Routing Table.

**to** **T0** Find the route entry used to reach an IP address or an exact network.

| T0 values   | Description                            |
|-------------|--|
| <A.B.C.D>   | An IPv4 address.                       |
| <A.B.C.D/M> | An IPv4 prefix: address and CIDR mask. |

## show ipv6-routes

```
vrouter> show ipv6-routes [vrf <string>] [protocol <identityref>] [table <uint32>] [to_
↵T0]
```

Show IPv6 routing table.

## Input Parameters

**vrf** <string> Specify the VRF.

**protocol** <identityref> Filter routes by protocol.

**table** <uint32> Non-main Kernel Routing Table.

**to** **T0** Find the route entry used to reach an IPv6 address or an exact network.

| T0 values    | Description                            |
|--------------|--|
| <X:X::X:X>   | An IPv6 address.                       |
| <X:X::X:X/M> | An IPv6 prefix: address and CIDR mask. |

## show mpls table

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show mpls table [<uint32>]
```

Show MPLS table information.

## Input Parameters

<uint32> LSP to display information about.

### show bgp

---

**Note:** requires a Turbo Router Network License.

---

```
vrouters> show bgp pbr ipset [set <string>] iptable [chain <string>] [vrf <string>] \
...      [vrfs] [summary] [neighbors] neighbor [id ID] community-list \
...      extcommunity-list as-path-access-list [name <string>] [route-map
↳<string>] \
...      ipv4 ip [VALUE] [bestpath] [multipath] prefix [value VALUE] [bestpath] \
...      [multipath] [longer-prefixes] [cidr-only] [statistics] [summary] \
...      [route-map <string>] flowspec ip [VALUE] [bestpath] [multipath] \
...      prefix [value VALUE] [bestpath] [multipath] [longer-prefixes] \
...      [detail] [cidr-only] [statistics] [summary] [route-map <string>] \
...      unicast neighbor [id ID] [prefix-counts] received [prefix-filter] \
...      [advertised-routes] [dampened-routes] [filtered-routes] [flap-
↳statistics] \
...      [received-routes] [routes] ip [VALUE] [bestpath] [multipath] \
...      prefix [value VALUE] [bestpath] [multipath] [longer-prefixes] \
...      [cidr-only] [statistics] [summary] [route-map <string>] multicast \
...      neighbor [id ID] [prefix-counts] [advertised-routes] [dampened-routes] \
...      [filtered-routes] [flap-statistics] [received-routes] [routes] \
...      ip [VALUE] [bestpath] [multipath] prefix [value VALUE] [bestpath] \
...      [multipath] [longer-prefixes] [cidr-only] [statistics] [summary] \
...      [route-map <string>] labeled-unicast neighbor [id ID] [advertised-
↳routes] \
...      [dampened-routes] [filtered-routes] [flap-statistics] [received-routes]_
↳\
...      [routes] ip [VALUE] [bestpath] [multipath] prefix [value VALUE] \
...      [bestpath] [multipath] [longer-prefixes] [cidr-only] [statistics] \
...      [summary] [route-map <string>] vpn [route-distinguisher ROUTE-
↳DISTINGUISHER] \
...      neighbor [id ID] [prefix-counts] [advertised-routes] [dampened-routes] \
...      [filtered-routes] [flap-statistics] [received-routes] [routes] \
...      ip [VALUE] [bestpath] [multipath] prefix [value VALUE] [bestpath] \
...      [multipath] [longer-prefixes] [cidr-only] [statistics] [summary] \
...      [route-map <string>] neighbor [id ID] [prefix-counts] received \
...      [prefix-filter] [advertised-routes] [dampened-routes] [filtered-routes]_
↳\
```

(continues on next page)

(continued from previous page)

```

...      [flap-statistics] [received-routes] [routes] [neighbors] ipv6 \
...      ip [value VALUE] [bestpath] [multipath] prefix [VALUE] [bestpath] \
...      [multipath] [longer-prefixes] [cidr-only] [statistics] [summary] \
...      [route-map <string>] flowspec ip [value VALUE] [bestpath] [multipath] \
...      prefix [VALUE] [bestpath] [multipath] [longer-prefixes] [detail] \
...      [cidr-only] [statistics] [summary] [route-map <string>] unicast \
...      neighbor [id ID] [prefix-counts] received [prefix-filter] [advertised-
↪routes] \
...      [dampened-routes] [filtered-routes] [flap-statistics] [received-routes]↵
↪\
...      [routes] ip [value VALUE] [bestpath] [multipath] prefix [VALUE] \
...      [bestpath] [multipath] [longer-prefixes] [cidr-only] [statistics] \
...      [summary] [route-map <string>] multicast neighbor [id ID] [prefix-
↪counts] \
...      [advertised-routes] [dampened-routes] [filtered-routes] [flap-
↪statistics] \
...      [received-routes] [routes] ip [value VALUE] [bestpath] [multipath] \
...      prefix [VALUE] [bestpath] [multipath] [longer-prefixes] [cidr-only] \
...      [statistics] [summary] [route-map <string>] labeled-unicast neighbor \
...      [id ID] [advertised-routes] [dampened-routes] [filtered-routes] \
...      [flap-statistics] [received-routes] [routes] ip [value VALUE] \
...      [bestpath] [multipath] prefix [VALUE] [bestpath] [multipath] \
...      [longer-prefixes] [cidr-only] [statistics] [summary] [route-map <string>
↪] \
...      vpn neighbor [id ID] [prefix-counts] [advertised-routes] [dampened-
↪routes] \
...      [filtered-routes] [flap-statistics] [received-routes] [routes] \
...      ip [value VALUE] [bestpath] [multipath] prefix [VALUE] [bestpath] \
...      [multipath] [longer-prefixes] [cidr-only] [statistics] [summary] \
...      [route-map <string>] neighbor [id ID] [prefix-counts] received \
...      [prefix-filter] [advertised-routes] [dampened-routes] [filtered-routes]↵
↪\
...      [flap-statistics] [received-routes] [routes] [neighbors] l2vpn \
...      evpn [vni VNI] [NET] [summary] [overlay] [tags] neighbor NEIGHBOR \
...      [advertised-routes] [routes] [route-distinguisher ROUTE-DISTINGUISHER] \
...      route [type TYPE] [detail]

```

Show BGP information.

### Input Parameters

**pbr ipset [set <string>] iptable [chain <string>]** Display information about PBR configured by BGP.

**ipset [set <string>]** Display information about PBR IPSETs configured by BGP.

**set <string>** Display information about this set.

**iptable [chain <string>]** Display information about PBR IPTables chainsa configured by BGP.

**chain <string>** Display information about this chain.

**vrf <string>** Specify the VRF.

**vrfs** Show BGP VRFs.

**summary** Summary of BGP neighbor status.

**neighbors** Display information about all BGP neighbors.

**neighbor [id ID]** Display information about one BGP neighbor.

**id ID** Display information about one BGP neighbor.

| ID values  | Description      |
|------------|------------------|
| <A.B.C.D>  | An IPv4 address. |
| <X:X::X:X> | An IPv6 address. |
| <string>   | No description.  |

**community-list** Display information about BGP community lists (standard and expanded).

**extcommunity-list** Display information about BGP extcommunity lists (standard and expanded).

**as-path-access-list [name <string>]** Display information about AS-path access lists.

**name <string>** Display information about a certain AS-Path access list.

**route-map <string>** Display information about this route map.

**ipv4 ip [VALUE] [bestpath] [multipath] prefix [value VALUE] [bestpath] [multipath] [longer-prefix]**  
 Display information about BGP IPv4.

**ip [VALUE] [bestpath] [multipath]** Display this address in the BGP routing table.

**VALUE** Display this address in the BGP routing table.

|       |                  |
|-------|------------------|
| VALUE | An IPv4 address. |
|-------|------------------|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**prefix** [**value** VALUE] [**bestpath**] [**multipath**] [**longer-prefixes**] Display this prefix in the BGP routing table.

**value** VALUE Display this prefix in the BGP routing table.

|       |  |
|-------|--|
| VALUE | An IPv4 prefix: address and CIDR mask. |
|-------|--|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**longer-prefixes** Display route and more specific routes.

**cidr-only** Display only routes with non-natural netmask.

**statistics** Display BGP RIB advertisement statistics.

**summary** Display summary of BGP neighbor status.

**route-map** <string> Display information about this route map.

**flowspec ip** [VALUE] [**bestpath**] [**multipath**] **prefix** [value VALUE] [**bestpath**] [**multipath**] [**longer-prefixes**] Display information for flowspec address family.

**ip** [VALUE] [**bestpath**] [**multipath**] Display this address in the BGP routing table.

**VALUE** Display this address in the BGP routing table.

|       |                  |
|-------|------------------|
| VALUE | An IPv4 address. |
|-------|------------------|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**prefix** [**value** VALUE] [**bestpath**] [**multipath**] [**longer-prefixes**] Display this prefix in the BGP routing table.

**value** VALUE Display this prefix in the BGP routing table.

|       |  |
|-------|--|
| VALUE | An IPv4 prefix: address and CIDR mask. |
|-------|--|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**longer-prefixes** Display route and more specific routes.

**detail** Display detailed information on flowspec entries.

**cidr-only** Display only routes with non-natural netmask.

**statistics** Display BGP RIB advertisement statistics.

**summary** Display summary of BGP neighbor status.

**route-map <string>** Display information about this route map.

**unicast neighbor [id ID] [prefix-counts] received [prefix-filter] [advertised-routes] [dampened-routes]**  
 Display information for unicast address family.

**neighbor [id ID] [prefix-counts] received [prefix-filter] [advertised-routes] [dampened-routes]**  
 Display information about one BGP neighbor.

**id ID** Display information about one BGP neighbor.

| ID values | Description      |
|-----------|------------------|
| <A.B.C.D> | An IPv4 address. |
| <string>  | No description.  |

**prefix-counts** Display detailed prefix count information.

**received [prefix-filter]** Display information received from a BGP neighbor.

**prefix-filter** Display the prefixlist filter.

**advertised-routes** Display the routes advertised to a BGP neighbor.

**dampened-routes** Display the dampened routes received from neighbor.

**filtered-routes** Display the filtered routes received from neighbor.

**flap-statistics** Display the flap statistics of the routes learned from neighbor.

**received-routes** Display the received routes from neighbor.

**routes** Display routes learned from neighbor.

**ip [VALUE] [bestpath] [multipath]** Display this address in the BGP routing table.

**VALUE** Display this address in the BGP routing table.

|       |                  |
|-------|------------------|
| VALUE | An IPv4 address. |
|-------|------------------|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**prefix [value VALUE] [bestpath] [multipath] [longer-prefixes]** Display this prefix in the BGP routing table.

**value VALUE** Display this prefix in the BGP routing table.

|       |  |
|-------|--|
| VALUE | An IPv4 prefix: address and CIDR mask. |
|-------|--|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**longer-prefixes** Display route and more specific routes.

**cidr-only** Display only routes with non-natural netmask.

**statistics** Display BGP RIB advertisement statistics.

**summary** Display summary of BGP neighbor status.

**route-map <string>** Display information about this route map.

**multicast neighbor [id ID] [prefix-counts] [advertised-routes] [dampened-routes] [filtered-routes]**  
 Display information for multicast address family.

**neighbor [id ID] [prefix-counts] [advertised-routes] [dampened-routes] [filtered-routes]**  
 Display information about one BGP neighbor.

**id ID** Display information about one BGP neighbor.

| ID values | Description      |
|-----------|------------------|
| <A.B.C.D> | An IPv4 address. |
| <string>  | No description.  |

**prefix-counts** Display detailed prefix count information.

**advertised-routes** Display the routes advertised to a BGP neighbor.

**dampened-routes** Display the dampened routes received from neighbor.

**filtered-routes** Display the filtered routes received from neighbor.

**flap-statistics** Display the flap statistics of the routes learned from neighbor.

**received-routes** Display the received routes from neighbor.

**routes** Display routes learned from neighbor.

**ip [VALUE] [bestpath] [multipath]** Display this address in the BGP routing table.

**VALUE** Display this address in the BGP routing table.

|       |                  |
|-------|------------------|
| VALUE | An IPv4 address. |
|-------|------------------|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**prefix [value VALUE] [bestpath] [multipath] [longer-prefixes]** Display this prefix in the BGP routing table.

**value VALUE** Display this prefix in the BGP routing table.

|       |  |
|-------|--|
| VALUE | An IPv4 prefix: address and CIDR mask. |
|-------|--|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**longer-prefixes** Display route and more specific routes.

**cidr-only** Display only routes with non-natural netmask.

**statistics** Display BGP RIB advertisement statistics.

**summary** Display summary of BGP neighbor status.

**route-map <string>** Display information about this route map.

**labeled-unicast neighbor [id ID] [advertised-routes] [dampened-routes] [filtered-routes] [flap-statistics]**  
 Display information for labeled unicast address family.

**neighbor [id ID] [advertised-routes] [dampened-routes] [filtered-routes] [flap-statistics]**  
 Display information about one BGP neighbor.

**id ID** Display information about one BGP neighbor.

| ID values | Description      |
|-----------|------------------|
| <A.B.C.D> | An IPv4 address. |
| <string>  | No description.  |

**advertised-routes** Display the routes advertised to a BGP neighbor.

**dampened-routes** Display the dampened routes received from neighbor.

**filtered-routes** Display the filtered routes received from neighbor.

**flap-statistics** Display the flap statistics of the routes learned from neighbor.

**received-routes** Display the received routes from neighbor.

**routes** Display routes learned from neighbor.

**ip [VALUE] [bestpath] [multipath]** Display this address in the BGP routing table.

**VALUE** Display this address in the BGP routing table.

|       |                  |
|-------|------------------|
| VALUE | An IPv4 address. |
|-------|------------------|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**prefix [value VALUE] [bestpath] [multipath] [longer-prefixes]** Display this prefix in the BGP routing table.

**value VALUE** Display this prefix in the BGP routing table.

|       |  |
|-------|--|
| VALUE | An IPv4 prefix: address and CIDR mask. |
|-------|--|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**longer-prefixes** Display route and more specific routes.

**cidr-only** Display only routes with non-natural netmask.

**statistics** Display BGP RIB advertisement statistics.

**summary** Display summary of BGP neighbor status.

**route-map <string>** Display information about this route map.

**vpn [route-distinguisher ROUTE-DISTINGUISHER] neighbor [id ID] [prefix-counts] [advertised-routes]**  
 Display information for VPN address family.

**route-distinguisher ROUTE-DISTINGUISHER** Display information for a route distinguisher.

| ROUTE-DISTINGUISHER values | Description   |
|----------------------------|---|
| <string>                   | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |
| <string>                   | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |

**neighbor [id ID] [prefix-counts] [advertised-routes] [dampened-routes] [filtered-routes]**  
 Display information about one BGP neighbor.

**id ID** Display information about one BGP neighbor.

| ID values | Description      |
|-----------|------------------|
| <A.B.C.D> | An IPv4 address. |
| <string>  | No description.  |

**prefix-counts** Display detailed prefix count information.

**advertised-routes** Display the routes advertised to a BGP neighbor.

**dampened-routes** Display the dampened routes received from neighbor.

**filtered-routes** Display the filtered routes received from neighbor.

**flap-statistics** Display the flap statistics of the routes learned from neighbor.

**received-routes** Display the received routes from neighbor.

**routes** Display routes learned from neighbor.

**ip** [VALUE] [bestpath] [multipath] Display this address in the BGP routing table.

**VALUE** Display this address in the BGP routing table.

|       |                  |
|-------|------------------|
| VALUE | An IPv4 address. |
|-------|------------------|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**prefix** [value VALUE] [bestpath] [multipath] [longer-prefixes] Display this prefix in the BGP routing table.

**value VALUE** Display this prefix in the BGP routing table.

|       |  |
|-------|--|
| VALUE | An IPv4 prefix: address and CIDR mask. |
|-------|--|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**longer-prefixes** Display route and more specific routes.

**cidr-only** Display only routes with non-natural netmask.

**statistics** Display BGP RIB advertisement statistics.

**summary** Display summary of BGP neighbor status.

**route-map** <string> Display information about this route map.

**neighbor** [id ID] [prefix-counts] received [prefix-filter] [advertised-routes] [dampened-routes] [filtered-routes] [flap-statistics] Display information about one BGP neighbor.

**id ID** Display information about one BGP neighbor.

| ID values | Description      |
|-----------|------------------|
| <A.B.C.D> | An IPv4 address. |
| <string>  | No description.  |

**prefix-counts** Display detailed prefix count information.

**received** [prefix-filter] Display information received from a BGP neighbor.

**prefix-filter** Display the prefixlist filter.

**advertised-routes** Display the routes advertised to a BGP neighbor.

**dampened-routes** Display the dampened routes received from neighbor.

**filtered-routes** Display the filtered routes received from neighbor.

**flap-statistics** Display the flap statistics of the routes learned from neighbor.

**received-routes** Display the received routes from neighbor.

**routes** Display routes learned from neighbor.

**neighbors** Display information about all BGP neighbors.

**ipv6 ip [value VALUE] [bestpath] [multipath] prefix [VALUE] [bestpath] [multipath] [longer-prefix]**  
Display information about BGP IPv6.

**ip [value VALUE] [bestpath] [multipath]** Display this address in the BGP routing table.

**value VALUE** Display this address in the BGP routing table.

|       |                  |
|-------|------------------|
| VALUE | An IPv6 address. |
|-------|------------------|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**prefix [VALUE] [bestpath] [multipath] [longer-prefixes]** Display this prefix in the BGP routing table.

**VALUE** Display this prefix in the BGP routing table.

|       |  |
|-------|--|
| VALUE | An IPv6 prefix: address and CIDR mask. |
|-------|--|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**longer-prefixes** Display route and more specific routes.

**cidr-only** Display only routes with non-natural netmask.

**statistics** Display BGP RIB advertisement statistics.

**summary** Display summary of BGP neighbor status.

**route-map <string>** Display information about this route map.

**flowspec ip [value VALUE] [bestpath] [multipath] prefix [VALUE] [bestpath] [multipath] [longer-prefixes]**  
Display information for flowspec address family.

**ip [value VALUE] [bestpath] [multipath]** Display this address in the BGP routing table.

**value VALUE** Display this address in the BGP routing table.

|       |                  |
|-------|------------------|
| VALUE | An IPv6 address. |
|-------|------------------|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**prefix [VALUE] [bestpath] [multipath] [longer-prefixes]** Display this prefix in the BGP routing table.

**VALUE** Display this prefix in the BGP routing table.

|       |  |
|-------|--|
| VALUE | An IPv6 prefix: address and CIDR mask. |
|-------|--|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**longer-prefixes** Display route and more specific routes.

**detail** Display detailed information on flowspec entries.

**cidr-only** Display only routes with non-natural netmask.

**statistics** Display BGP RIB advertisement statistics.

**summary** Display summary of BGP neighbor status.

**route-map <string>** Display information about this route map.

**unicast neighbor [id ID] [prefix-counts] received [prefix-filter] [advertised-routes] [dampened-routes]** Display information for unicast address family.

**neighbor [id ID] [prefix-counts] received [prefix-filter] [advertised-routes] [dampened-routes]** Display information about one BGP neighbor.

**id ID** Display information about one BGP neighbor.

| ID values  | Description      |
|------------|------------------|
| <X:X::X:X> | An IPv6 address. |
| <string>   | No description.  |

**prefix-counts** Display detailed prefix count information.

**received [prefix-filter]** Display information received from a BGP neighbor.

**prefix-filter** Display the prefixlist filter.

**advertised-routes** Display the routes advertised to a BGP neighbor.

**dampened-routes** Display the dampened routes received from neighbor.

**filtered-routes** Display the filtered routes received from neighbor.

**flap-statistics** Display the flap statistics of the routes learned from neighbor.

**received-routes** Display the received routes from neighbor.

**routes** Display routes learned from neighbor.

**ip [value VALUE] [bestpath] [multipath]** Display this address in the BGP routing table.

**value VALUE** Display this address in the BGP routing table.

|       |                  |
|-------|------------------|
| VALUE | An IPv6 address. |
|-------|------------------|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**prefix [VALUE] [bestpath] [multipath] [longer-prefixes]** Display this prefix in the BGP routing table.

**VALUE** Display this prefix in the BGP routing table.

|       |  |
|-------|--|
| VALUE | An IPv6 prefix: address and CIDR mask. |
|-------|--|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**longer-prefixes** Display route and more specific routes.

**cidr-only** Display only routes with non-natural netmask.

**statistics** Display BGP RIB advertisement statistics.

**summary** Display summary of BGP neighbor status.

**route-map <string>** Display information about this route map.

**multicast neighbor [id ID] [prefix-counts] [advertised-routes] [dampened-routes] [filtered-routes]** Display information for multicast address family.

**neighbor [id ID] [prefix-counts] [advertised-routes] [dampened-routes] [filtered-routes]** Display information about one BGP neighbor.

**id ID** Display information about one BGP neighbor.

| ID values  | Description      |
|------------|------------------|
| <X:X::X:X> | An IPv6 address. |
| <string>   | No description.  |

**prefix-counts** Display detailed prefix count information.

**advertised-routes** Display the routes advertised to a BGP neighbor.

**dampened-routes** Display the dampened routes received from neighbor.

**filtered-routes** Display the filtered routes received from neighbor.

**flap-statistics** Display the flap statistics of the routes learned from neighbor.

**received-routes** Display the received routes from neighbor.

**routes** Display routes learned from neighbor.

**ip [value VALUE] [bestpath] [multipath]** Display this address in the BGP routing table.

**value VALUE** Display this address in the BGP routing table.

|       |                  |
|-------|------------------|
| VALUE | An IPv6 address. |
|-------|------------------|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**prefix [VALUE] [bestpath] [multipath] [longer-prefixes]** Display this prefix in the BGP routing table.

**VALUE** Display this prefix in the BGP routing table.

|       |  |
|-------|--|
| VALUE | An IPv6 prefix: address and CIDR mask. |
|-------|--|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**longer-prefixes** Display route and more specific routes.

**cidr-only** Display only routes with non-natural netmask.

**statistics** Display BGP RIB advertisement statistics.

**summary** Display summary of BGP neighbor status.

**route-map <string>** Display information about this route map.

**labeled-unicast neighbor [id ID] [advertised-routes] [dampened-routes] [filtered-routes] [flap-statistics]** Display information for labeled unicast address family.

**neighbor [id ID] [advertised-routes] [dampened-routes] [filtered-routes] [flap-statistics]** Display information about one BGP neighbor.

**id ID** Display information about one BGP neighbor.

| ID values  | Description      |
|------------|------------------|
| <X:X::X:X> | An IPv6 address. |
| <string>   | No description.  |

**advertised-routes** Display the routes advertised to a BGP neighbor.

**dampened-routes** Display the dampened routes received from neighbor.

**filtered-routes** Display the filtered routes received from neighbor.

**flap-statistics** Display the flap statistics of the routes learned from neighbor.

**received-routes** Display the received routes from neighbor.

**routes** Display routes learned from neighbor.

**ip [value VALUE] [bestpath] [multipath]** Display this address in the BGP routing table.

**value VALUE** Display this address in the BGP routing table.

|       |                  |
|-------|------------------|
| VALUE | An IPv6 address. |
|-------|------------------|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**prefix [VALUE] [bestpath] [multipath] [longer-prefixes]** Display this prefix in the BGP routing table.

**VALUE** Display this prefix in the BGP routing table.

|       |  |
|-------|--|
| VALUE | An IPv6 prefix: address and CIDR mask. |
|-------|--|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**longer-prefixes** Display route and more specific routes.

**cidr-only** Display only routes with non-natural netmask.

**statistics** Display BGP RIB advertisement statistics.

**summary** Display summary of BGP neighbor status.

**route-map <string>** Display information about this route map.

**vpn neighbor [id ID] [prefix-counts] [advertised-routes] [dampened-routes] [filtered-routes]**  
Display information for VPN address family.

**neighbor [id ID] [prefix-counts] [advertised-routes] [dampened-routes] [filtered-routes]**  
Display information about one BGP neighbor.

**id ID** Display information about one BGP neighbor.

| ID values  | Description      |
|------------|------------------|
| <X:X::X:X> | An IPv6 address. |
| <string>   | No description.  |

**prefix-counts** Display detailed prefix count information.

**advertised-routes** Display the routes advertised to a BGP neighbor.

**dampened-routes** Display the dampened routes received from neighbor.

**filtered-routes** Display the filtered routes received from neighbor.

**flap-statistics** Display the flap statistics of the routes learned from neighbor.

**received-routes** Display the received routes from neighbor.

**routes** Display routes learned from neighbor.

**ip [value VALUE] [bestpath] [multipath]** Display this address in the BGP routing table.

**value VALUE** Display this address in the BGP routing table.

|       |                  |
|-------|------------------|
| VALUE | An IPv6 address. |
|-------|------------------|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**prefix [VALUE] [bestpath] [multipath] [longer-prefixes]** Display this prefix in the BGP routing table.

**VALUE** Display this prefix in the BGP routing table.

|       |  |
|-------|--|
| VALUE | An IPv6 prefix: address and CIDR mask. |
|-------|--|

**bestpath** Display only the best path.

**multipath** Display only multipaths.

**longer-prefixes** Display route and more specific routes.

**cidr-only** Display only routes with non-natural netmask.

**statistics** Display BGP RIB advertisement statistics.

**summary** Display summary of BGP neighbor status.

**route-map <string>** Display information about this route map.

**neighbor [id ID] [prefix-counts] received [prefix-filter] [advertised-routes] [dampened-routes]** Display information about one BGP neighbor.

**id ID** Display information about one BGP neighbor.

| ID values  | Description      |
|------------|------------------|
| <X:X::X:X> | An IPv6 address. |
| <string>   | No description.  |

**prefix-counts** Display detailed prefix count information.

**received [prefix-filter]** Display information received from a BGP neighbor.

**prefix-filter** Display the prefixlist filter.

- advertised-routes** Display the routes advertised to a BGP neighbor.
- dampened-routes** Display the dampened routes received from neighbor.
- filtered-routes** Display the filtered routes received from neighbor.
- flap-statistics** Display the flap statistics of the routes learned from neighbor.
- received-routes** Display the received routes from neighbor.
- routes** Display routes learned from neighbor.

**neighbors** Display information about all BGP neighbors.

**l2vpn evpn [vni VNI] [NET] [summary] [overlay] [tags] neighbor NEIGHBOR [advertised-routes] [routes]** Display Layer 2 Virtual Private Network information.

**evpn [vni VNI] [NET] [summary] [overlay] [tags] neighbor NEIGHBOR [advertised-routes] [routes]** Display Ethernet Virtual Private Network information.

**vni VNI** Display VNI information.

|     |   |
|-----|---|
| VNI | Type definition representing VXLAN Segment ID / VXLAN Network Identifier value. |
|-----|---|

**NET** Network in the BGP routing table to display.

| NET values  | Description                            |
|-------------|--|
| <A.B.C.D>   | An IPv4 address.                       |
| <A.B.C.D/M> | An IPv4 prefix: address and CIDR mask. |

**summary** Summary of BGP neighbor status.

**overlay** Display BGP Overlay Information for prefixes.

**tags** Display BGP tags for prefixes.

**neighbor NEIGHBOR [advertised-routes] [routes]** Detailed information on TCP and BGP neighbor connections.

**NEIGHBOR (mandatory)** Neighbor to display information about.

| NEIGHBOR values | Description      |
|-----------------|------------------|
| <A.B.C.D>       | An IPv4 address. |
| <X:X::X:X>      | An IPv6 address. |

**advertised-routes** Display the routes advertised to a BGP neighbor.

**routes** Display routes learned from neighbor.

**route-distinguisher ROUTE-DISTINGUISHER** Display information for a route distinguisher.

| ROUTE-DIST values | Description   |
|-------------------|---|
| <string>          | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |
| <string>          | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |

**route [type TYPE] [detail]** Detailed information BPG L2VPN EVPN routes.

**type TYPE** Specify route type.

| TYPE values | Description               |
|-------------|---------------------------|
| macip       | MAC-IP (Type-2) route.    |
| multicast   | Multicast (Type-3) route. |
| prefix      | Prefix (Type-5) route.    |

**detail** Display detail information.

### show bgp rpki cache-connection

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show bgp rpki cache-connection [vrf VRF]
```

Show which RPKI cache servers have a connection.

### Input Parameters

**vrf VRF** Specify the VRF.

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

## show bgp rpki cache-server

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show bgp rpki cache-server [vrf VRF]
```

Show RPKI configured cache server.

### Input Parameters

**vrf VRF** Specify the VRF.

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

## show bgp rpki prefix-table

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show bgp rpki prefix-table [vrf VRF] [PREFIX] [as AS]
```

Show validated prefixes which were received from RPKI Cache.

### Input Parameters

**vrf VRF** Specify the VRF.

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

**PREFIX** Lookup by IPv4/IPv6 prefix.

| PREFIX values | Description                            |
|---------------|--|
| <A.B.C.D/M>   | An IPv4 prefix: address and CIDR mask. |
| <X:X::X:X/M>  | An IPv6 prefix: address and CIDR mask. |

**as AS** Lookup by AS number.

|    |   |
|----|---|
| AS | A numeric identifier for an autonomous system (AS). An AS is a single domain, under common administrative control, which forms a unit of routing policy. Autonomous systems can be assigned a 2-byte identifier, or a 4-byte identifier which may have public or private scope. Private ASNs are assigned from dedicated ranges. Public ASNs are assigned from ranges allocated by IANA to the regional internet registries (RIRs). |
|----|---|

### show evpn

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show evpn [arp-cache] [mac] vni VNI [detail]
```

Show EVPN information.

### Input Parameters

**arp-cache** Show ARP and ND cache information.

**mac** Show MAC addresses information.

**vni VNI (mandatory)** Show EVPN information about a specific VNI or all.

| VNI values | Description   |
|------------|---|
| all        | Show all VNIs.  |
| <uint32>   | Type definition representing VXLAN Segment ID / VXLAN Network Identifier value. |

**detail** Detail information on each VNI.

### show ospf

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show ospf [vrf <string>] [vrfs] [route] database [default] [max-age] router \
... [ADDRESS] [advertising-router ADVERTISING-ROUTER] asbr-summary \
... [ADDRESS] [advertising-router ADVERTISING-ROUTER] external [ADDRESS] \
... [advertising-router ADVERTISING-ROUTER] network [ADDRESS] [advertising-
←router ADVERTISING-ROUTER] \
```

(continues on next page)

(continued from previous page)

```

...      nssa-external [ADDRESS] [advertising-router ADVERTISING-ROUTER] \
...      opaque-area [ADDRESS] [advertising-router ADVERTISING-ROUTER] \
...      opaque-link [ADDRESS] [advertising-router ADVERTISING-ROUTER] \
...      opaque-as [ADDRESS] [advertising-router ADVERTISING-ROUTER] summary \
...      [ADDRESS] [advertising-router ADVERTISING-ROUTER] neighbor [ADDRESS] \
...      [IFNAME] [detail] interface [traffic] [NAME]
    
```

Show OSPF information.

### Input Parameters

**vrf** <string> Specify the VRF.

**vrf**s Available VRFs.

**route** OSPF routing table.

**database** [default] [max-age] router [ADDRESS] [advertising-router ADVERTISING-ROUTER] asbr-summary Database summary.

**default** Database summary.

**max-age** Database maximum age.

**router** [ADDRESS] [advertising-router ADVERTISING-ROUTER] Database Router link states.

**ADDRESS** The router address.

| ADDRESS values | Description                  |
|----------------|------------------------------|
| <A.B.C.D>      | An IPv4 address.             |
| self-originate | Self-originated link states. |

**advertising-router** ADVERTISING-ROUTER The advertising router address.

|                    |                  |
|--------------------|------------------|
| ADVERTISING-ROUTER | An IPv4 address. |
|--------------------|------------------|

**asbr-summary** [ADDRESS] [advertising-router ADVERTISING-ROUTER] Database ASBR summary link states.

**ADDRESS** The router address.

| ADDRESS values | Description                  |
|----------------|------------------------------|
| <A.B.C.D>      | An IPv4 address.             |
| self-originate | Self-originated link states. |

**advertising-router ADVERTISING-ROUTER** The advertising router address.

|                    |                  |
|--------------------|------------------|
| ADVERTISING-ROUTER | An IPv4 address. |
|--------------------|------------------|

**external [ADDRESS] [advertising-router ADVERTISING-ROUTER]** Database External link states.

**ADDRESS** The router address.

| ADDRESS values | Description                  |
|----------------|------------------------------|
| <A.B.C.D>      | An IPv4 address.             |
| self-originate | Self-originated link states. |

**advertising-router ADVERTISING-ROUTER** The advertising router address.

|                    |                  |
|--------------------|------------------|
| ADVERTISING-ROUTER | An IPv4 address. |
|--------------------|------------------|

**network [ADDRESS] [advertising-router ADVERTISING-ROUTER]** Database Network link states.

**ADDRESS** The router address.

| ADDRESS values | Description                  |
|----------------|------------------------------|
| <A.B.C.D>      | An IPv4 address.             |
| self-originate | Self-originated link states. |

**advertising-router ADVERTISING-ROUTER** The advertising router address.

|                    |                  |
|--------------------|------------------|
| ADVERTISING-ROUTER | An IPv4 address. |
|--------------------|------------------|

**nssa-external [ADDRESS] [advertising-router ADVERTISING-ROUTER]** Database NSSA external link states.

**ADDRESS** The router address.

| ADDRESS values | Description                  |
|----------------|------------------------------|
| <A.B.C.D>      | An IPv4 address.             |
| self-originate | Self-originated link states. |

**advertising-router ADVERTISING-ROUTER** The advertising router address.

|                    |                  |
|--------------------|------------------|
| ADVERTISING-ROUTER | An IPv4 address. |
|--------------------|------------------|

**opaque-area [ADDRESS] [advertising-router ADVERTISING-ROUTER]** Database Opaque link state area.

**ADDRESS** The router address.

| ADDRESS values | Description                  |
|----------------|------------------------------|
| <A.B.C.D>      | An IPv4 address.             |
| self-originate | Self-originated link states. |

**advertising-router ADVERTISING-ROUTER** The advertising router address.

|                    |                  |
|--------------------|------------------|
| ADVERTISING-ROUTER | An IPv4 address. |
|--------------------|------------------|

**opaque-link [ADDRESS] [advertising-router ADVERTISING-ROUTER]** Database Opaque link states.

**ADDRESS** The router address.

| ADDRESS values | Description                  |
|----------------|------------------------------|
| <A.B.C.D>      | An IPv4 address.             |
| self-originate | Self-originated link states. |

**advertising-router ADVERTISING-ROUTER** The advertising router address.

|                    |                  |
|--------------------|------------------|
| ADVERTISING-ROUTER | An IPv4 address. |
|--------------------|------------------|

**opaque-as [ADDRESS] [advertising-router ADVERTISING-ROUTER]** Database Opaque AS link states.

**ADDRESS** The router address.

| ADDRESS values | Description                  |
|----------------|------------------------------|
| <A.B.C.D>      | An IPv4 address.             |
| self-originate | Self-originated link states. |

**advertising-router ADVERTISING-ROUTER** The advertising router address.

|                    |                  |
|--------------------|------------------|
| ADVERTISING-ROUTER | An IPv4 address. |
|--------------------|------------------|

**summary [ADDRESS] [advertising-router ADVERTISING-ROUTER]** Database Summary link states.

**ADDRESS** The router address.

| ADDRESS values | Description                  |
|----------------|------------------------------|
| <A.B.C.D>      | An IPv4 address.             |
| self-originate | Self-originated link states. |

**advertising-router** **ADVERTISING-ROUTER** The advertising router address.

|                    |                  |
|--------------------|------------------|
| ADVERTISING-ROUTER | An IPv4 address. |
|--------------------|------------------|

**neighbor** [**ADDRESS**] [**IFNAME**] [**detail**] Neighbors information.

**ADDRESS** Neighbor ID.

|         |                  |
|---------|------------------|
| ADDRESS | An IPv4 address. |
|---------|------------------|

**IFNAME** The interface name.

|        |                    |
|--------|--------------------|
| IFNAME | An interface name. |
|--------|--------------------|

**detail** Show detailed neighbor's information.

**interface** [**traffic**] [**NAME**] Interface information.

**traffic** Interface traffic information.

**NAME** The interface name. If not specified, show all interfaces.

|      |                    |
|------|--------------------|
| NAME | An interface name. |
|------|--------------------|

## show rip

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show rip [vrf <string>] [status]
```

Show RIP information.

## Input Parameters

**vrf** <string> Specify the VRF.

**status** Show RIP status.

## show ospf6

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show ospf6 [vrf <string>] route [DESTINATION] database [default] [router] \
...           [neighbor] interface [NAME]
```

Show OSPFv3 information.

### Input Parameters

**vrf <string>** Specify the VRF.

**route [DESTINATION]** OSPFv3 routing table.

**DESTINATION** The route destination.

| DESTINATION values | Description                            |
|--------------------|--|
| <X:X::X:X>         | An IPv6 address.                       |
| <X:X::X:X/M>       | An IPv6 prefix: address and CIDR mask. |
| detail             | Detailed information.                  |
| external-1         | Display Type-1 External routes.        |
| external-2         | Display Type-2 External routes.        |
| inter-area         | Display Inter-Area routes.             |
| intra-area         | Display Intra-Area routes.             |
| summary            | Route table summary.                   |

**database [default] [router]** Database summary.

**default** Database summary.

**router** Database Router link states.

**neighbor** Neighbor list.

**interface [NAME]** Interface information.

**NAME** The interface name. If not specified, show all interfaces.

|      |                    |
|------|--------------------|
| NAME | An interface name. |
|------|--------------------|

## show ripng

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show ripng [status]
```

Show RIPng information.

### Input Parameters

**status** Show RIPng status.

## show mpls ldp

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show mpls ldp discovery [detail] [interface] [capabilities] neighbor [LSR-ID] \
↪ \
... [capabilities] [detail] binding [PREFIX] [longer-prefixes] [local-label
↪ <uint32>] \
... [remote-label <uint32>] [neighbor NEIGHBOR] [detail] [ipv4] [ipv6]
```

Show MPLS LDP information.

### Input Parameters

**discovery [detail]** Discovery Hello Information.

**detail** Show detailed information.

**interface** Interface information.

**capabilities** Display neighbor capability information.

**neighbor [LSR-ID] [capabilities] [detail]** Neighbor information.

**LSR-ID** OSPF routing table.

|        |                  |
|--------|------------------|
| LSR-ID | An IPv4 address. |
|--------|------------------|

**capabilities** Display neighbor capability information.

**detail** Show detailed information.

**binding** [**PREFIX**] [**longer-prefixes**] [**local-label** <uint32>] [**remote-label** <uint32>] [**neighbor** NEIGHBOR] Label Information Base (LIB) information.

**PREFIX** Destination prefix.

| PREFIX values | Description                            |
|---------------|--|
| <A.B.C.D/M>   | An IPv4 prefix: address and CIDR mask. |
| <X:X::X:X/M>  | An IPv6 prefix: address and CIDR mask. |

**longer-prefixes** Include longer matches.

**local-label** <uint32> Locally assigned label value.

**remote-label** <uint32> Match remotely assigned label values.

**neighbor** NEIGHBOR Display labels from LDP neighbor.

|          |                  |
|----------|------------------|
| NEIGHBOR | An IPv4 address. |
|----------|------------------|

**detail** Show detailed information.

**ipv4** IPv4 Address Family.

**ipv6** IPv6 Address Family.

### show bfd

---

**Note:** requires a Turbo Router Network License.

---

```
vrouters> show bfd [vrf VRF] [address ADDRESS] [HOP-TYPE] [source SOURCE] [interface_
↪INTERFACE] \
... [counters]
```

Show BFD information.

### Input Parameters

**vrf** VRF Specify the VRF.

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

**address ADDRESS** IP address of the peer.

| ADDRESS values | Description      |
|----------------|------------------|
| <A.B.C.D>      | An IPv4 address. |
| <X:X::X:X>     | An IPv6 address. |

**HOP-TYPE** Show single or multi hop session.

| HOP-TYPE values | Description              |
|-----------------|--------------------------|
| single-hop      | Show single-hop session. |
| multi-hop       | Show multi-hop session.  |

**source SOURCE** Local IP address.

| SOURCE values | Description                  |
|---------------|------------------------------|
| <A.B.C.D>     | An IPv4 address.             |
| <X:X::X:X>    | An IPv6 address.             |
| any           | Accept any source addresses. |

**interface INTERFACE** Interface used to contact peer.

|           |                    |
|-----------|--------------------|
| INTERFACE | An interface name. |
|-----------|--------------------|

**counters** Show BFD session counters information.

### show path-monitoring

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show path-monitoring [vrf VRF] [address ADDRESS] [operational]
```

Show path monitoring information.

## Input Parameters

**vrf** **VRF** Specify the VRF.

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

**address** **ADDRESS** IP address of the peer.

| ADDRESS values | Description      |
|----------------|------------------|
| <A.B.C.D>      | An IPv4 address. |
| <X:X::X:X>     | An IPv6 address. |

**operational** Show session operational information.

## show nhrp

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show nhrp [vrf <string>] [cache] [nhs] [opennhrp] [shortcut] [default]
```

Show NHRP IPv4 information.

## Input Parameters

**vrf** <string> Specify the VRF.

**cache** NHRP forwarding cache information.

**nhs** NHRP Next hop server information.

**opennhrp** NHRP opennhrpctl style cache dump.

**shortcut** NHRP shortcut information.

**default** NHRP default information.

## show nhrp6

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show nhrp6 [vrf <string>] [cache] [nhs] [opennhrp] [shortcut] [default]
```

Show NHRP IPv6 information.

### Input Parameters

**vrf** <string> Specify the VRF.

**cache** NHRP forwarding cache information.

**nhs** NHRP Next hop server information.

**opennhrp** NHRP opennhrpctl style cache dump.

**shortcut** NHRP shortcut information.

**default** NHRP default information.

## show license

```
vrouter> show license
```

Show license information.

## show boot-params

```
vrouter> show boot-params
```

Show boot parameters. Image must be installed on disk.

### Output Data

**current** Current boot parameters.

**intel-iommu true|false** Enable intel iommu driver. Control intel\_iommu=on|off kernel option.

**iommu-allow-unsafe-interrupts true|false** Enable PCI passthrough on hardware that does not support interrupt remapping, when VM are trusted. Control vfiommu\_type1.allow\_unsafe\_interrupts=0|1 kernel option.

**ixgbe-allow-unsupported-sfp true|false** Bypass SFPs types restrictions on Intel ixgbe NICs. Control `ixgbe.allow_unsupported_sfp=0|1` kernel option.

**isolate-cpus ISOLATE-CPUS** Isolate cpus from kernel threads, rcu callbacks, and reduce the scheduler ticks. A good value for this parameter is the fast path coremask.

| ISOLATE-CPUS values | Description   |
|---------------------|---|
| <cores-list>        | A comma-separated list of cores or core ranges. Example: '1,4-7,10-12'. |
| none                | Unset the coremask.   |

**next** Next boot parameters.

**intel-iommu true|false** Enable intel iommu driver. Control `intel_iommu=on|off` kernel option.

**iommu-allow-unsafe-interrupts true|false** Enable PCI passthrough on hardware that does not support interrupt remapping, when VM are trusted. Control `vfiio_iommu_type1.allow_unsafe_interrupts=0|1` kernel option.

**ixgbe-allow-unsupported-sfp true|false** Bypass SFPs types restrictions on Intel ixgbe NICs. Control `ixgbe.allow_unsupported_sfp=0|1` kernel option.

**isolate-cpus ISOLATE-CPUS** Isolate cpus from kernel threads, rcu callbacks, and reduce the scheduler ticks. A good value for this parameter is the fast path coremask.

| ISOLATE-CPUS values | Description   |
|---------------------|---|
| <cores-list>        | A comma-separated list of cores or core ranges. Example: '1,4-7,10-12'. |
| none                | Unset the coremask.   |

### show log

```
vrouter> show log [max-lines <uint16>] [service <identityref>] [vrf <string>] \
↳[facility FACILITY] \
... level [EQUAL] [greater-or-equal GREATER-OR-EQUAL] not [LEVEL]
```

Print log.

### Input Parameters

**max-lines <uint16>** Log max lines.

**service <identityref>** Filter logs by service.

**vrf <string>** Filter logs by VRF.

**facility FACILITY** Filter logs by facility.

| FACILITY values | Description  |
|-----------------|--|
| kernel          | Filter kernel messages.                                    |
| mail            | Filter mail system messages.                               |
| news            | Filter network news subsystem messages.                    |
| user            | Filter random user-level messages.                         |
| auth            | Filter security/authorization messages.                    |
| authpriv        | Filter security/authorization messages (private).          |
| cron            | Filter clock daemon messages.                              |
| daemon          | Filter system daemons messages.                            |
| line-printer    | Filter line printer subsystem messages.                    |
| FTP             | Filter FTP daemon messages.                                |
| syslog          | Filter messages generated internally by the syslog daemon. |
| uucp            | Filter UUCP subsystem messages.                            |
| local0          | Filter messages from local0.                               |
| local1          | Filter messages from local1.                               |
| local2          | Filter messages from local2.                               |
| local3          | Filter messages from local3.                               |
| local4          | Filter messages from local4.                               |
| local5          | Filter messages from local5.                               |
| local6          | Filter messages from local6.                               |
| local7          | Filter messages from local7.                               |
| any             | Filter messages from any facilities.                       |

**level [EQUAL] [greater-or-equal GREATER-OR-EQUAL] not [LEVEL]** Filter logs by level.

**EQUAL** Select levels to show.

| EQUAL values | Description                           |
|--------------|---------------------------------------|
| emergency    | System is unusable.                   |
| alert        | Action must be taken immediately.     |
| critical     | Critical conditions.                  |
| error        | Error conditions.                     |
| warning      | Warning conditions.                   |
| notice       | Normal but significant condition.     |
| info         | Informational messages.               |
| debug        | Debug-level messages.                 |
| any          | Show all messages from this facility. |

**greater-or-equal GREATER-OR-EQUAL** Filter messages with a greater or equal level than the selected one.

| GREATER-OR-EQUAL values | Description                       |
|-------------------------|-----------------------------------|
| emergency               | System is unusable.               |
| alert                   | Action must be taken immediately. |
| critical                | Critical conditions.              |
| error                   | Error conditions.                 |
| warning                 | Warning conditions.               |
| notice                  | Normal but significant condition. |
| info                    | Informational messages.           |
| debug                   | Debug-level messages.             |

**not** [LEVEL] Select levels to not show.

**LEVEL** Do not show messages with this level.

| LEVEL values | Description                       |
|--------------|-----------------------------------|
| emergency    | System is unusable.               |
| alert        | Action must be taken immediately. |
| critical     | Critical conditions.              |
| error        | Error conditions.                 |
| warning      | Warning conditions.               |
| notice       | Normal but significant condition. |
| info         | Informational messages.           |
| debug        | Debug-level messages.             |

## show ntp

```
vrouter> show ntp [vrf <string>] [details]
```

Show NTP information.

### Input Parameters

**vrf** <string> VRF to look into.

**details** Show per server details.

## show dhcp-server

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show dhcp-server [vrf <string>]
```

Show DHCP server leases.

### Input Parameters

**vrf <string>** Specify the VRF.

## show contracks

```
vrouter> show contracks [vrf <string>] [family FAMILY] [protocol PROTOCOL]
```

Show contracks.

### Input Parameters

**vrf <string>** The VRF in which to show the contracks.

**family FAMILY** Display only this layer 3 family.

| FAMILY values | Description     |
|---------------|-----------------|
| ipv4          | IPv4 only.      |
| ipv6          | IPv6 only.      |
| <string>      | No description. |

**protocol PROTOCOL** Display only this layer 4 protocol.

| PROTOCOL values | Description     |
|-----------------|-----------------|
| tcp             | TCP only.       |
| udp             | UDP only.       |
| <string>        | No description. |

## show product

```
vrouter> show product [name] [version]
```

Show the product name and version.

### Input Parameters

**name** Display the product name.

**version** Display the product version.

## show fast-path cpu-usage

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show fast-path cpu-usage
```

Show the fast path CPU usage.

## show fast-path table-usage

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show fast-path table-usage
```

Show the fastpath table usage.

## show fast-path statistics

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show fast-path statistics [all] [TYPE]
```

Show fast-path statistics.

## Input Parameters

**all** Show all fast-path services statistics.

**TYPE** Show fast-path services statistics.

| TYPE values    | Description                                       |
|----------------|---|
| ipv4           | Show ipv4 fast-path service statistics.           |
| ipv6           | Show ipv6 fast-path service statistics.           |
| vxlan          | Show vxlan fast-path service statistics.          |
| vlan           | Show vlan fast-path service statistics.           |
| bridge         | Show bridge fast-path service statistics.         |
| lag            | Show lag fast-path service statistics.            |
| gre            | Show GRE fast-path service statistics.            |
| mpls           | Show MPLS fast-path service statistics.           |
| global         | Show global fast-path service statistics.         |
| interface      | Show interface fast-path service statistics.      |
| exception      | Show exception fast-path service statistics.      |
| qos-sched      | Show QoS scheduler fast-path service statistics.  |
| qos-rate-limit | Show QoS rate limit fast-path service statistics. |

## Output Data

**ipv4** IPv4 service statistics.

**ip-forwarded-datagrams** <uint64> Number of IP packets forwarded (IpForwDatagrams).

**ip-in-delivered** <uint64> Number of IP packets delivered to user-protocols (IpInDelivers).

**ip-in-received** <uint64> Number of IP packets received (IpInReceives).

**ip-in-truncated-packets** <uint64> Number of IP packets discarded due to a truncate IP header (IpInTruncatedPkts).

**ip-in-address-errors** <uint64> Number of IP packets discarded due to invalid IP address (IpInAddrErrors).

**ip-in-header-errors** <uint64> Number of IP packets discarded due to errors in header (IpInHdrErrors).

**ip-fragment-created** <uint64> Number of IP fragment packets created on fragmentation processing (IpFragCreates).

**ip-fragment-ok** <uint64> Number of IP fragment packets sent successfully (IpFragOKs).

**ip-fragment-failures** <uint64> Number of IP packets discarded due to failures during fragmentation processing (IpFragFails).

**ip-reassembly-ok** <uint64> Number of IP packets successfully reassembled (IpReasmOKs).

- ip-reassembly-required** <uint64> Number of IP fragments packets submitted to reassembly processing (IpReasmReqds).
- ip-reassembly-exceptions** <uint64> Number of IP fragment packets sent in exception path (IpReasmExceptions).
- ip-reassembly-failures** <uint64> Number of IP packets discarded due to failures during reassembly processing (IpReasmFails).
- ip-reassembly-dropped-duplicate** <uint64> Number of IP packets dropped during reassembly considered as duplicate (IpReasmDroppedDuplicate).
- ip-reassembly-dropped-session-complete** <uint64> Number of IP packets dropped during reassembly because the session is complete (IpReasmDroppedSessionComplete).
- ip-reassembly-dropped-session-full** <uint64> Number of IP packets dropped during reassembly because the session is already full (IpReasmDroppedSessionAlreadyFull).
- ip-reassembly-error-header-encapsulation** <uint64> Number of IP packets discarded during reassembly due to header encapsulation error (IpReasmErrorHeaderEncap).
- ip-reassembly-error-ip-option-unsupported** <uint64> Number of IP packets discarded during reassembly due to unsupported IP option (IpReasmErrorIPOptionUnsupported).
- ip-reassembly-error-last-already-received** <uint64> Number of IP packets discarded during reassembly due to receive twice the last fragment (IpReasmErrorLastAlreadyReceived).
- ip-reassembly-error-offset-too-large** <uint64> Number of IP packets discarded during reassembly with offset due to an offset too big (IpReasmErrorOffsetTooLarge).
- ip-reassembly-error-overlap-next** <uint64> Number of IP packets discarded during reassembly due to receive overlapping fragment with next one (IpReasmErrorOverlapNext).
- ip-reassembly-error-overlap-previous** <uint64> Number of IP packets discarded during reassembly due to receive overlapping fragment with previous one (IpReasmErrorOverlapPrevious).
- ip-reassembly-error-packet-too-short** <uint64> Number of IP packets discarded during reassembly due to reception of a too short fragment (IpReasmErrorPacketTooShort).
- ip-reassembly-error-queue-allocation** <uint64> Number of IP packets discarded during reassembly due to reassembly queue allocation failure (IpReasmErrorQueueAlloc).
- ip-reassembly-error-queue-full** <uint64> Number of IP packets discarded during reassembly due to reassembly queue full (too many fragments have been received) (IpReasmErrorQueueFull).
- ip-reassembly-error-size-exceed** <uint64> Number of IP packets discarded during reassembly due to total received bytes greater than the maximal authorized value (65535) (IpReasmErrorSizeExceed).
- ip-reassembly-error-size-overflow** <uint64> Number of IP packets discarded during reassembly due to total received bytes greater than the expected value (IpReasmErrorSizeOverflow).
- ip-reassembly-error-too-many-segments** <uint64> Number of IP packets discarded during reassembly due to too many segments in IP packets (IpReasmErrorTooManySegments).

**ip-reassembly-timeout <uint64>** Number of IP packets discarded due to timeout in reassembly processing (IpReasmTimeout).

**ip-checksum-errors <uint64>** Number of IP packets discarded due to an invalid checksum (IpCsumErrors).

**ip-dropped-blackhole <uint64>** Number of IP packets discarded due to matching blackhole route (IpDroppedBlackhole).

**ip-dropped-filtering <uint64>** Number of IP packets discarded by filtering processing (IpDroppedNetfilter).

**ip-dropped-forwarding <uint64>** Number of IP packets discarded due to forwarding being disabled (IpDroppedForwarding).

**ip-dropped-invalid-interface <uint64>** Number of IP packets discarded due to invalid outgoing interface (IpDroppedInvalidInterface).

**ip-dropped-ipsec <uint64>** Number of IP packets discarded by IPsec processing (IpDroppedIPsec).

**ip-dropped-no-arp <uint64>** Number of IP packets discarded due to missing ARP resolution (IpDroppedNoArp).

**ip-dropped-no-memory <uint64>** Number of IP packets discarded due to memory allocation errors (IpDroppedNoMemory).

**ip-dropped-out-operative <uint64>** Number of IP packets discarded because the outgoing interface is down (IPDroppedOutOperative).

**ip-dropped-route-exception <uint64>** Number of IP packets sent to exception due to specific route (IpDroppedRouteException).

**ip-nhrp-packet <uint64>** Number of IP NHRP packets (IpNhrpPacket).

**ip-nhrp-error-send <uint64>** Number of discarded sent IP NHRP packets (IpNhrpErrorSend).

**ipv6** IPv6 service statistics.

**ip6-forwarded-datagrams <uint64>** Number of IPv6 packets forwarded (IpForwDatagrams).

**ip6-in-delivered <uint64>** Number of IPv6 packets delivered to user-protocols (IpInDelivers).

**ip6-in-received <uint64>** Number of IPv6 packets received (IpInReceives).

**ip6-in-truncated-packets <uint64>** Number of IPv6 packets discarded due to a truncate IP header (IpInTruncatedPkts).

**ip6-in-address-errors <uint64>** Number of IPv6 packets discarded due to invalid IPv6 address (IpInAddrErrors).

**ip6-in-header-errors <uint64>** Number of IPv6 packets discarded due to errors in header (IpInHdrErrors).

**ip6-fragment-created <uint64>** Number of IPv6 fragment packets created on fragmentation processing (IpFragCreates).

- ip6-fragment-ok <uint64>** Number of IPv6 fragment packets sent successfully (IpFragOKs).
- ip6-fragment-failures <uint64>** Number of IPv6 packets discarded due to failures during fragmentation processing (IpFragFails).
- ip6-fragment-reassembly-exceptions <uint64>** Number of IP fragment packets sent in exception path.
- ip6-reassembly-ok <uint64>** Number of IPv6 packets successfully reassembled (IpReasmOKs).
- ip6-reassembly-required <uint64>** Number of IPv6 fragments packets submitted to reassembly processing (IpReasmReqds).
- ip6-reassembly-exceptions <uint64>** Number of IPv6 fragment packets sent in exception path (IpReasmExceptions).
- ip6-reassembly-failures <uint64>** Number of IPv6 packets discarded due to failures during reassembly processing (IpReasmFails).
- ip6-reassembly-dropped-session-complete <uint64>** Number of IPv6 packets dropped during reassembly because the session is complete (IpReasmDroppedSessionComplete).
- ip6-reassembly-dropped-session-full <uint64>** Number of IPv6 packets dropped during reassembly because the session is already full (IpReasmDroppedSessionAlreadyFull).
- ip6-reassembly-error-fragment-header <uint64>** Number of IPv6 packets discarded during reassembly due to header reading error (IpReasmErrorFragmentHeader).
- ip6-reassembly-error-header-encapsulation <uint64>** Number of IPv6 packets discarded during reassembly due to header encapsulation error (IpReasmErrorHeaderEncap).
- ip6-reassembly-error-ip6-option-too-large <uint64>** Number of IPv6 packets discarded during reassembly due to IPv6 option too large (IpReasmErrorIPOptionTooLarge).
- ip6-reassembly-error-last-already-received <uint64>** Number of IPv6 packets discarded during reassembly due to receive twice the last fragment (IpReasmErrorLastAlreadyReceived).
- ip6-reassembly-error-offset-too-large <uint64>** Number of IPv6 packets discarded during reassembly with offset due to an offset too big (IpReasmErrorOffsetTooLarge).
- ip6-reassembly-error-overlap-next <uint64>** Number of IPv6 packets discarded during reassembly due to receive overlapping fragment with next one (IpReasmErrorOverlapNext).
- ip6-reassembly-error-overlap-previous <uint64>** Number of IPv6 packets discarded during reassembly due to receive overlapping fragment with previous one (IpReasmErrorOverlapPrevious).
- ip6-reassembly-error-packet-too-short <uint64>** Number of IPv6 packets discarded during reassembly due to reception of a too short fragment (IpReasmErrorPacketTooShort).
- ip6-reassembly-error-queue-allocation <uint64>** Number of IPv6 packets discarded during reassembly due to reassembly queue allocation failure (IpReasmErrorQueueAlloc).
- ip6-reassembly-error-queue-full <uint64>** Number of IPv6 packets discarded during reassembly due to reassembly queue full (too many fragments have been received) (IpReasmErrorQueueFull).

**ip6-reassembly-error-size-exceed <uint64>** Number of IPv6 packets discarded during reassembly due to total received bytes greater than the maximal authorized value (65535) (IpReasmErrorSizeExceed).

**ip6-reassembly-error-size-overflow <uint64>** Number of IPv6 packets discarded during reassembly due to total received bytes greater than the expected value (IpReasmErrorSizeOverflow).

**ip6-reassembly-error-too-many-segments <uint64>** Number of IPv6 packets discarded during reassembly due to too many segments in IP packets (IpReasmErrorTooManySegments).

**ip6-reassembly-timeout <uint64>** Number of IPv6 packets discarded due to timeout in reassembly processing (IpReasmTimeout).

**ip6-dropped-blackhole <uint64>** Number of IPv6 packets discarded due to matching blackhole route (IpDroppedBlackhole).

**ip6-dropped-filtering <uint64>** Number of IPv6 packets discarded by filtering processing (IpDroppedNetfilter).

**ip6-dropped-forwarding <uint64>** Number of IPv6 packets discarded due to forwarding being disabled (IpDroppedForwarding).

**ip6-dropped-invalid-interface <uint64>** Number of IPv6 packets discarded due to invalid outgoing interface (IpDroppedInvalidInterface).

**ip6-dropped-ipsec <uint64>** Number of IPv6 packets discarded by IPsec processing (IpDroppedIPsec).

**ip6-dropped-no-arp <uint64>** Number of IPv6 packets discarded due to missing ARP resolution (IpDroppedNoArp).

**ip6-dropped-no-memory <uint64>** Number of IPv6 packets discarded due to memory allocation errors (IpDroppedNoMemory).

**ip6-dropped-out-operative <uint64>** Number of IPv6 packets discarded because the outgoing interface is down (IPDroppedOutOperative).

**ip6-dropped-route-exception <uint64>** Number of IPv6 packets sent to exception due to specific route (IpDroppedRouteException).

**ip6-nhrp-packet <uint64>** Number of IPv6 NHRP packets (IpNhrpPacket).

**ip6-nhrp-error-send <uint64>** Number of discarded sent IPv6 NHRP packets (IpNhrpErrorSend).

**vxlan** VXLAN service statistics.

**vxlan-dropped-header-too-short <uint64>** Number of input packets dropped in VXLAN due to a VXLAN header too short (VxlanDroppedHeaderTooShort).

**vxlan-dropped-in-operative <uint64>** Number of input packets dropped in VXLAN because the incoming interface is down (VxlanDroppedInOperative).

**vxlan-dropped-invalid-ip-family <uint64>** Number of output packets dropped in VXLAN due to a failure to get the VXLAN header (VxlanDroppedInvalidIpFamily).

- vxlan-dropped-invalid-ipv4-checksum** <uint64> Number of input packets dropped in IPv4 VXLAN due to an invalid checksum (VxlanDroppedInvalidIPv4Csum).
- vxlan-dropped-invalid-ipv4-header** <uint64> Number of input packets dropped in IPv4 VXLAN due to a failure to get the VXLAN header (VxlanDroppedInvalidIPv4Header).
- vxlan-dropped-invalid-ipv6-checksum** <uint64> Number of input packets dropped in IPv6 VXLAN due to an invalid checksum (VxlanDroppedInvalidIPv6Csum).
- vxlan-dropped-invalid-ipv6-header** <uint64> Number of input packets dropped in IPv6 VXLAN due to a failure to get the VXLAN header (VxlanDroppedInvalidIPv6Header).
- vxlan-dropped-ipv4-no-destination** <uint64> Number of output packets dropped in IPv4 VXLAN due to a null destination address (VxlanDroppedIPv4NoDst).
- vxlan-dropped-ipv6-no-destination** <uint64> Number of output packets dropped in IPv6 VXLAN due to a null destination address (VxlanDroppedIPv6NoDst).
- vxlan-dropped-ovs-no-destination** <uint64> Number of output packets dropped in OVS VXLAN due to a null destination address (VxlanDroppedOvsNoDst).
- vxlan-dropped-prepend-ipv4-failure** <uint64> Number of output packets dropped in IPv4 VXLAN due to add IP header (VxlanDroppedPrependIPv4Failure).
- vxlan-dropped-prepend-ipv6-failure** <uint64> Number of output packets dropped in IPv6 VXLAN due to add IP header (VxlanDroppedPrependIPv6Failure).
- vxlan-dropped-prepend-ovs-failure** <uint64> Number of output packets dropped in OVS VXLAN due to add IP header (VxlanDroppedPrependOvsFailure).
- vxlan-dropped-unknown-iface** <uint64> Number of input packets dropped in VXLAN due to an invalid interface (VxlanDroppedUnknownIface).
- vxlan-dropped-unknown-vni** <uint64> Number of input packets dropped in VXLAN due to an invalid VNI (VxlanDroppedUnknownVNI).
- vxlan-exception-i-flag-not-set** <uint64> Number of input packets sent to exception by VXLAN due a I flags not set (see rfc 7348) (VxlanExceptionIFlagNotSet).
- vxlan-exception-ipv4-mtu-exceeded** <uint64> Number of output packets sent to exception by IPv4 VXLAN due a MTU exceeded the authorized value (VxlanExceptionIPv4MtuExceeded).
- vxlan-exception-ipv4-no-multicast-source** <uint64> Number of output packets sent to exception by IPv4 VXLAN due to no valid src address found for a multicast packet (VxlanException-IPv4NoMcastSrc).
- vxlan-exception-ipv4-route** <uint64> Number of output packets sent to exception by IPv4 VXLAN due to specific route (VxlanExceptionIPv4Route).
- vxlan-exception-ipv6-mtu-exceeded** <uint64> Number of output packets sent to exception by IPv6 VXLAN due a MTU exceeded the authorized value (VxlanExceptionIPv6MtuExceeded).

**vxlan-exception-ipv6-no-multicast-source <uint64>** Number of output packets sent to exception by IPv6 VXLAN due to no valid src address found for a multicast packet (VxlanExceptionIPv6NoMcastSrc).

**vxlan-exception-ipv6-route <uint64>** Number of output packets sent to exception by IPv6 VXLAN due to specific route (VxlanExceptionIPv6Route).

**vxlan-exception-no-input-fdb <uint64>** Number of input packets sent to exception by VXLAN due to no valid fdb found (VxlanExceptionNoInputFdb).

**vxlan-exception-no-output-fdb <uint64>** Number of output packets sent to exception by VXLAN due to no valid fdb found (VxlanExceptionNoOutputFdb).

**vxlan-exception-no-remote <uint64>** Number of output packets sent to exception by VXLAN due to no remote found (VxlanExceptionNoRemote).

**vxlan-exception-ovs-mtu-exceeded <uint64>** Number of output packets sent to exception by Ovs VXLAN due a MTU exceeded the authorized value (VxlanExceptionOvsMtuExceeded).

**vxlan-exception-ovs-route <uint64>** Number of output packets sent to exception by Ovs VXLAN due to specific route (VxlanExceptionOvsRoute).

**vxlan-exception-too-many-flags <uint64>** Number of input packets sent to exception by VXLAN due to a presence of an unsupported flag (neither I and G ones, see rfc 7348) (VxlanExceptionTooManyFlags).

**vxlan-fdb-forwarding-duplicate-error <uint64>** Number of failure to duplicate a packet for fdb forwarding (VxlanFdbForwDuplicateError).

**vlan** VLAN service statistics.

**vlan-dropped-in-operative <uint64>** Number of input packets dropped in VLAN because the incoming interface is down (VlanDroppedInOperative).

**vlan-dropped-input-unknown-interface <uint64>** Number of input packets dropped in VLAN due to unknown interface (VlanDroppedInputUnknownIf).

**vlan-dropped-invalid-tag <uint64>** Number of input packets dropped in VLAN due to an invalid tag (VlanDroppedInvalidTag).

**vlan-dropped-out-operative <uint64>** Number of output packets dropped in VLAN because the outgoing interface is down (VlanDroppedOutOperative).

**vlan-dropped-prepend-failure <uint64>** Number of output packets dropped in VLAN due to a failure to add VLAN tag (VlanDroppedPrependFailure).

**vlan-output-unknown-interface <uint64>** Number of output packets with unknown interface (VlanOutputUnknownIf).

**vlan-unknown-tag <uint64>** Number of packets with unknown VLAN tag (VlanUnknownTag).

**bridge** Bridge service statistics.

**l2-forwarded-frames <uint64>** Number of packets forwarded at layer 2 (bridging processing) (L2ForwFrames).

**bridge-dropped-forwarding-invalid <uint64>** Number of output packets dropped in bridge due to forbidden forwarding (forwarding disable or originating port) (BridgeDroppedFwdInvalid).

**bridge-dropped-input-lookup-error <uint64>** Number of input packets dropped in bridge due to a lookup error (BridgeDroppedInputLookupError).

**bridge-dropped-invalid-output-port <uint64>** Number of output packets dropped in bridge because output port index is invalid (BridgeDroppedInvalidOutPort).

**bridge-dropped-invalid-source <uint64>** Number of input packets dropped in bridge due to invalid mac source (BridgeDroppedInvalidSrc).

**bridge-dropped-invalid-state <uint64>** Number of input packets dropped in bridge due to invalid state (not learning or forwarding) of the bridge (BridgeDroppedInvalidState).

**bridge-dropped-learning <uint64>** Number of output packets dropped in bridge while it is in learning state (BridgeDroppedLearning).

**bridge-dropped-mtu-exceeded <uint64>** Number of output packets dropped in bridge due to MTU greater than the authorized one (BridgeDroppedMtuExceeded).

**bridge-dropped-no-output-port <uint64>** Number of output packets dropped in bridge due to no valid output (BridgeDroppedNoOutputPort).

**bridge-dropped-output-lookup-error <uint64>** Number of output packets dropped in bridge due to a lookup error (BridgeDroppedOutputLookupError).

**bridge-dropped-out-operative <uint64>** Number of output packets dropped in bridge because the outgoing interface is down (BridgeDroppedOutOperative).

**bridge-dropped-output-unknown <uint64>** Number of output packets dropped in bridge due to an unknown output (BridgeDroppedOutputUnknown).

**bridge-dropped-pause-frame <uint64>** Number of input packets dropped in bridge because it is a pause frame (BridgeDroppedPauseFrame).

**bridge-dropped-unknown-interface <uint64>** Number of input packets dropped in bridge due to an invalid interface (BridgeDroppedUnknownIface).

**bridge-fdb-synchronization-error <uint64>** Number of packets dropped in bridge due to fdb synchronization error (BridgeFdbSyncError).

**lag** Lag service statistics.

**lag-dropped-inactive-port <uint64>** Number of LAG dropped inactive ports (LagDroppedInactivePort).

**gre** GRE service statistics.

**gre-dropped-init-gre-ipv4-header-failure <uint64>** Number of output packets dropped in GRE due to a failure to add the GRE header (GREdroppedInitGreIPv4HeaderFailure).

**gre-dropped-init-gre-ipv6-header-failure <uint64>** Number of output packets dropped in GRE6 due to a failure to add the GRE header (GREdroppedInitGreIPv6HeaderFailure).

- gre-dropped-init-ipv4-header-failure <uint64>** Number of output packets dropped in GRE due to a failure to add the IPv4 header (GREDroppedInitIPv4HeaderFailure).
- gre-dropped-init-ipv6-header-failure <uint64>** Number of output packets dropped in GRE due to a failure to add the IPv6 header (GREDroppedInitIPv6HeaderFailure).
- gre-dropped-in-operative <uint64>** Number of input packets dropped in GRE because the ingoing interface is down (GREDroppedInOperative).
- gre-dropped-missing-checksum <uint64>** Number of input packets dropped in GRE due to a missing checksum (GREDroppedMissingChecksum).
- gre-dropped-out-operative <uint64>** Number of output packets dropped in GRE because the outgoing interface is down (GREDroppedOutOperative).
- gre-dropped-parse-ipv4-header-failure <uint64>** Number of input packets dropped in GRE due to failure to parse IPv4 header (GREDroppedParseIPv4HeaderFailure).
- gre-dropped-parse-ipv6-header-failure <uint64>** Number of input packets dropped in GRE due to failure to parse IPv6 header (GREDroppedParseIPv6HeaderFailure).
- gre-dropped-pullup-ipv4-header-failure <uint64>** Number of input packets dropped in IPv4 GRE due to pullup failure on gre header (GREDroppedPullupIPv4HeaderFailure).
- gre-dropped-pullup-ipv6-header-failure <uint64>** Number of input packets dropped in IPv6 GRE due to pullup failure on gre header (GREDroppedPullupIPv6HeaderFailure).
- gre-dropped-unexpected-checksum <uint64>** Number of input packets dropped in GRE due to an unexpected checksum (GREDroppedUnexpectedChecksum).
- gre-dropped-wrong-checksum <uint64>** Number of input packets dropped in GRE due to an incorrect checksum (GREDroppedWrongChecksum).
- gre-exception-input-unsupported-protocol <uint64>** Number of input packets sent to exception by GRE due to unsupported GRE protocol (GREExceptionInputUnsupportedProtocol).
- gre-exception-ipv4-route <uint64>** Number of output packets sent to exception by GRE due to specific route (for IPv4 packet) (GREExceptionIPv4Route).
- gre-exception-ipv4-source-select-failed <uint64>** Number of output packets sent to exception by GRE due to no src address can be set (for IPv4 packet) (GREExceptionIPv4SourceSelectFailed).
- gre-exception-ipv6-route <uint64>** Number of output packets sent to exception by GRE due to specific route (for IPv6 packet) (GREExceptionIPv6Route).
- gre-exception-output-unsupported-protocol <uint64>** Number of output packets sent to exception by GRE due to unsupported GRE protocol (GREExceptionOutputUnsupportedProtocol).
- gre-exception-unknown-iface <uint64>** Number of output packets sent to exception by GRE due to an invalid GRE interface id (GREExceptionUnknownIface).
- gre-exception-unsupported-ethernet-type <uint64>** Number of output packets sent to exception by GRE due to unsupported ethernet type (GREExceptionUnsupportedEtherType).

**gre-invalid-header <uint64>** Number of input packets not managed by GRE due to routing flags set (see rfc 1701) or version number different to 0. The packet can be dropped or sent to exception later in other fast path processing part (GREInvalidHeader).

**gre-protocol-not-supported <uint64>** Number of input packets not supported by GRE (GREProtocolNotSupported).

**gretap-dropped-out-operative <uint64>** Number of output packets dropped in GRETAP because the outgoing interface is down (GRETAPDroppedOutOperative).

**gretap-exception-unknown-iface <uint64>** Number of output packets sent to exception by GRETAP due to an invalid GRE interface id (GRETAPExceptionUnknownIface).

**mpls** MPLS service statistics.

**mpls-forwarding <uint64>** Number of forwarding packets in MPLS (MplsForwarding).

**mpls-input <uint64>** Number of input packets in MPLS (MplsInput).

**mpls-no-route <uint64>** Number of packets in MPLS with no route (MplsNoRoute).

**mpls-push <uint64>** Number of push packets in MPLS (MplsPush).

**mpls-in-header-errors <uint64>** Number of packets in MPLS discarded due to errors in header (MplsInHdrErrors).

**mpls-received-dropped <uint64>** Number of received dropped packets in MPLS (MplsRxDrop).

**mpls-dropped-invalid-interface <uint64>** Number of packets dropped in MPLS due to invalid outgoing interface (MplsDroppedInvalidInterface).

**mpls-dropped-mtu <uint64>** Number of packets dropped in MPLS due to MTU greater than the authorized one (MplsDroppedMtu).

**mpls-dropped-no-memory <uint64>** Number of packets dropped in MPLS due to memory allocation errors (MplsDroppedNoMem).

**mpls-dropped-no-neighbor <uint64>** Number of dropped packets in MPLS due to no neighbor is found (MplsDroppedNoNeigh).

**mpls-dropped-ttl-exceed <uint64>** Number of packets dropped in MPLS due to ttl exceeded (MplsDroppedTtlExceed).

**global** Global service statistics.

**fast-path-dropped <uint64>** Number of packets dropped by fast path (fp\_dropped).

**fast-path-dropped-arp <uint64>** Number of packets dropped by fast path in ARP (fp\_dropped\_arp).

**fast-path-dropped-bonding <uint64>** Number of packets dropped by fast path in bonding (fp\_dropped\_bonding).

**fast-path-dropped-bridge <uint64>** Number of packets dropped by fast path in bridge (fp\_dropped\_bridge).

- fast-path-dropped-ebtables** <uint64> Number of packets dropped by fast path in layer 2 filtering (fp\_dropped\_ebtables).
- fast-path-dropped-ethernet** <uint64> Number of packets dropped by fast path at the generic ethernet layer (fp\_dropped\_ether).
- fast-path-dropped-exception** <uint64> Number of packets dropped by fast path in exception path (fp\_dropped\_excpc).
- fast-path-dropped-exception-loop** <uint64> Number of packets dropped by fast path due to exception loop (fp\_dropped\_excloop).
- fast-path-dropped-filtering** <uint64> Number of packets dropped by fast path in IPv4 filtering (fp\_dropped\_netfilter).
- fast-path-dropped-filtering-ipv6** <uint64> Number of packets dropped by fast path in IPv6 filtering (fp\_dropped\_netfilter6).
- fast-path-dropped-gre** <uint64> Number of packets dropped by fast path in GRE (fp\_dropped\_gre).
- fast-path-dropped-ip** <uint64> Number of packets dropped by fast path in generic IPv4 (fp\_dropped\_ip).
- fast-path-dropped-ipsec** <uint64> Number of packets dropped by fast path in IPv4 IPsec (fp\_dropped\_ipsec).
- fast-path-dropped-ipsec-ipv6** <uint64> Number of packets dropped by fast path in IPv6 IPsec (fp\_dropped\_ipsec6).
- fast-path-dropped-ipv6** <uint64> Number of packets dropped by fast path in generic IPv6 (fp\_dropped\_ipv6).
- fast-path-dropped-macvlan** <uint64> Number of packets dropped by fast path in MACVLAN (fp\_dropped\_macvlan).
- fast-path-dropped-mpls** <uint64> Number of packets dropped by fast path in MultiProtocol Label Switching (fp\_dropped\_mpls).
- fast-path-dropped-multicast** <uint64> Number of packets dropped by fast path in IPv4 multicast (fp\_dropped\_mcast).
- fast-path-dropped-multicast-ipv6** <uint64> Number of packets dropped by fast path in IPv6 multicast (fp\_dropped\_mcast6).
- fast-path-dropped-npf** <uint64> Number of packets dropped by fast path in Network Address Translation and Carrier-grade NAT (fp\_dropped\_npf).
- fast-path-dropped-ovs** <uint64> Number of packets dropped by fast path in Open vSwitch (fp\_dropped\_ovs).
- fast-path-dropped-plugins** <uint64> Number of packets dropped by fast path in plugin (fp\_dropped\_plugins).
- fast-path-dropped-qos** <uint64> Number of packets dropped by fast path in QoS (fp\_dropped\_qos).

**fast-path-dropped-reassembly** <uint64> Number of packets dropped by fast path in IPv4 reassembly (fp\_dropped\_reasm).

**fast-path-dropped-reassembly-ipv6** <uint64> Number of packets dropped by fast path in IPv6 reassembly (fp\_dropped\_reasm6).

**fast-path-dropped-system** <uint64> Number of packets dropped by fast path in internal processing (fp\_dropped\_system).

**fast-path-dropped-tc** <uint64> Number of packets dropped by fast path in generic traffic conditioning (fp\_dropped\_tc).

**fast-path-dropped-tc-erl** <uint64> Number of packets dropped by fast path in traffic conditioning by exception rate limitation (fp\_dropped\_tc\_erl).

**fast-path-dropped-tunnel** <uint64> Number of packets dropped by fast path in IPinIP tunnel (fp\_dropped\_tunnel).

**fast-path-dropped-vethernet** <uint64> Number of packets dropped by fast path in vEthernet (fp\_dropped\_veth).

**fast-path-dropped-vlan** <uint64> Number of packets dropped by fast path in VLAN (fp\_dropped\_vlan).

**fast-path-dropped-vxlan** <uint64> Number of packets dropped by fast path in VXLAN (fp\_dropped\_vxlan).

**fast-path-missing-ipsec-license** <uint64> Number of packets dropped in fast path due to missing ipsec license (fp\_missing\_ipsec\_license).

**fast-path-missing-product-license** <uint64> Number of packets dropped in fast path due to missing product license (fp\_missing\_product\_license).

**interface** Interface statistics.

**name** <string> Interface name.

**accelerated** true|false True if the interface is managed by the fast-path, else false.

**input-bytes** <uint64> The number of input received bytes (ifs\_abytes).

**input-errors** <uint64> The number of input received errors (ifs\_ierrors).

**input-last-error** <uint64> The number of input received last errors (ifs\_ilasterror).

**input-multicasts** <uint64> The number of input received multicast packets (ifs\_imcasts).

**input-no-mbuf** <uint64> The number of input packets dropped because no mbuf was available (ifs\_inombuf).

**input-packets** <uint64> The number of input received packets (ifs\_ipackets).

**missed-input-packets** <uint64> The number of missed input packets (ifs\_imissed).

**multicasts** <uint64> The number of multicasts (ifs\_mcasts).

**output-bytes** <uint64> The number of output sent bytes (ifs\_obytes).

**output-errors** <uint64> The number of output sent errors (ifs\_oerrors).

**output-packets** <uint64> The number of output sent packets (ifs\_opackets).

**exception** Exception service statistics.

**exception-by-module** Exceptions by module statistics.

**fast-path-exception-bonding** <uint64> Number of packets send in exception in bonding (fp\_exception\_bonding).

**fast-path-exception-bridge** <uint64> Number of packets send in exception in bridge (fp\_exception\_bridge).

**fast-path-exception-ebtables** <uint64> Number of packets send in exception in layer 2 filtering (fp\_exception\_ebtables).

**fast-path-exception-ethernet** <uint64> Number of packets send in exception in generic layer 2 (fp\_exception\_ether).

**fast-path-exception-filtering** <uint64> Number of packets send in exception in IPv4 filtering (fp\_exception\_netfilter).

**fast-path-exception-filtering-ipv6** <uint64> Number of packets send in exception in IPv6 filtering (fp\_exception\_netfilter6).

**fast-path-exception-gre** <uint64> Number of packets send in exception in GRE (fp\_exception\_gre).

**fast-path-exception-ifnet** <uint64> Number of packets send in exception by a virtual interface (fp\_exception\_ifnet).

**fast-path-exception-ip** <uint64> Number of packets send in exception in generic IPv4 (fp\_exception\_ip).

**fast-path-exception-ipsec** <uint64> Number of packets send in exception in IPv4 IPsec (fp\_exception\_ipsec).

**fast-path-exception-ipsec-ipv6** <uint64> Number of packets send in exception in IPv6 IPsec (fp\_exception\_ipsec6).

**fast-path-exception-ipv6** <uint64> Number of packets send in exception in generic IPv6 (fp\_exception\_ipv6).

**fast-path-exception-macvlan** <uint64> Number of packets send in exception in MACVLAN (fp\_exception\_macvlan).

**fast-path-exception-mpls** <uint64> Number of packets send in exception in MPLS (fp\_exception\_mpls).

**fast-path-exception-npf** <uint64> Number of packets send in exception in NPF (fp\_exception\_npf).

**fast-path-exception-reassembly** <uint64> Number of packets send in exception in IPv4 re-assembly (fp\_exception\_reass).

- fast-path-exception-sflow <uint64>** Number of packets sent in exception in sflow (fp\_exception\_sflow).
- fast-path-exception-syslog <uint64>** Number of packets send in exception for logging (for system without syslog) (fp\_exception\_syslog).
- fast-path-exception-tap <uint64>** Number of packets send in exception in eBPF (Enhanced Berkeley Packet Filtering), typically when there is a tcpdump or sflow (fp\_exception\_tap).
- fast-path-exception-tunnel <uint64>** Number of packets send in exception in IPinIP tunnel (fp\_exception\_tunnel).
- fast-path-exception-unknown-ifnet <uint64>** Number of packets sent in exception due to unknown ifnet (fp\_exception\_unknown\_ifnet).
- fast-path-exception-vethernet <uint64>** Number of packets sent in exception in vEthernet (fp\_exception\_veth).
- fast-path-exception-vlan <uint64>** Number of packets sent in exception in VLAN (fp\_exception\_vlan).
- fast-path-exception-vxlan <uint64>** Number of packets send in exception in VXLAN (fp\_exception\_vxlan).
- exception-dropped-fp-to-linux-add-mark-failure <uint64>** Number of exception packets to Linux dropped due to a tag addition failure (ExcpDroppedFpToLinuxAddMarkFailure).
- exception-dropped-fp-to-linux-fptun-failure <uint64>** Number of ExcpDroppedFpToLinuxFptunFailure exceptions (ExcpDroppedFpToLinuxFptunFailure).
- exception-dropped-fp-to-linux-no-ipv4-route-local <uint64>** Number of exception packets to Linux dropped due to a failure to find the IPv4 route (ExcpDroppedFpToLinuxNoIPv4RouteLocal).
- exception-dropped-fp-to-linux-no-ipv6-route-local <uint64>** Number of exception packets to Linux dropped due to a failure to find the IPv6 route (ExcpDroppedFpToLinuxNoIPv6RouteLocal).
- exception-dropped-fp-to-linux-prepend-failure <uint64>** Number of ExcpDroppedFpToLinuxPrependFailure exceptions (ExcpDroppedFpToLinuxPrependFailure).
- exception-dropped-fp-to-linux-prepend-failure-detailed** Detailed exceptions of the packets dropped from fast-path to linux with prepend failure.
- exception-dropped-fp-to-linux-ecmp-prepend-failure <uint64>** Number of ExcpDroppedFpToLinuxEcmpPrependFailure exceptions (ExcpDroppedFpToLinuxEcmpPrependFailure).
- exception-dropped-fp-to-linux-ecmp6-prepend-failure <uint64>** Number of ExcpDroppedFpToLinuxEcmp6PrependFailure exceptions (ExcpDroppedFpToLinuxEcmp6PrependFailure).
- exception-dropped-fp-to-linux-eth-fptun-prepend-failure <uint64>** Number of ExcpDroppedFpToLinuxEthFptunPrependFailure exceptions (ExcpDroppedFpToLinuxEthFptunPrependFailure).

- exception-dropped-fp-to-linux-eth-prepend-failure <uint64>** Number of ExcpDroppedFpToLinuxEthPrependFailure exceptions (ExcpDroppedFpToLinuxEthPrependFailure).
- exception-dropped-fp-to-linux-ipsec-prepend-failure <uint64>** Number of ExcpDroppedFpToLinuxIPsecPrependFailure exceptions (ExcpDroppedFpToLinuxIPsecPrependFailure).
- exception-dropped-fp-to-linux-restore-failure <uint64>** Number of ExcpDroppedFpToLinuxRestoreFailure exceptions (ExcpDroppedFpToLinuxRestoreFailure).
- exception-dropped-fp-to-linux-tuple-prepend-failure <uint64>** Number of ExcpDroppedFpToLinuxTuplePrependFailure exceptions (ExcpDroppedFpToLinuxTuplePrependFailure).
- exception-dropped-invalid-mtag <uint64>** Number of ExcpDroppedInvalidMtag exceptions (ExcpDroppedInvalidMtag).
- exception-dropped-linux-to-fp-generic-command-failure <uint64>** Number of packets from Linux dropped due to a FPTUN internal error (ExcpDroppedLinuxToFpGenericCommandFailure).
- exception-dropped-linux-to-fp-invalid-port-id <uint64>** Number of packets from Linux dropped due to a reception of a FPTUN message on an unexpected port (ExcpDroppedLinuxToFpInvalidPortId).
- exception-dropped-linux-to-fp-invalid-version <uint64>** Number of packets from Linux dropped due to an invalid FPTUN version (ExcpDroppedLinuxToFpInvalidVersion).
- exception-dropped-linux-to-fp-ipv4-pullup-failure <uint64>** Number of packets from Linux dropped due to a failure when getting the IPv4 header of the FPTUN message (ExcpDroppedLinuxToFpIPv4PullupFailure).
- exception-dropped-linux-to-fp-ipv6-pullup-failure <uint64>** Number of packets from Linux dropped due to a failure when getting the IPv6 header of the FPTUN message (ExcpDroppedLinuxToFpIPv6PullupFailure).
- exception-dropped-linux-to-fp-msg-too-short <uint64>** Number of packets from Linux dropped due to an incomplete FPTUN message (ExcpDroppedLinuxToFpMsgTooShort).
- exception-dropped-linux-to-fp-no-output-function <uint64>** Number of packets from Linux dropped because no TX function has been registered (ExcpDroppedLinuxToFpNoOutputFunction).
- exception-dropped-linux-to-fp-other-host <uint64>** Number of packets from Linux dropped due to a reception of a FPTUN message marked as PACKET\_OTHERHOST (ExcpDroppedLinuxToFpOtherHost).
- exception-dropped-linux-to-fp-out-operative <uint64>** Number of packets from Linux dropped because the destination interface is down (ExcpDroppedLinuxToFpOutOperative).
- exception-dropped-linux-to-fp-tproxy-failure <uint64>** Number of ExcpDroppedLinuxToFpTproxyFailure exceptions (ExcpDroppedLinuxToFpTproxyFailure).
- exception-dropped-linux-to-fp-unknown-command <uint64>** Number of packets from Linux dropped due to an invalid FPTUN command (ExcpDroppedLinuxToFpUnknownCommand).

**exception-dropped-linux-to-fp-unknown-interface-uid <uint64>** Number of packets from Linux dropped due to an invalid interface id for FPTUN (ExcpDroppedLinuxToFpUnknownIfUid).

**exception-dropped-no-contrack <uint64>** Number of ExcpDroppedNoContrack exceptions (ExcpDroppedNoContrack).

**local-basic-exceptions <uint64>** The number of local basic exceptions (LocalBasicExceptions).

**local-exception-class** Local exception class statistics.

**fptun-exception-ecmp-ndisc-needed <uint64>** The number of FPTUN\_EXC\_ECMP\_NDISC\_NEEDED exceptions (FPTUN\_EXC\_ECMP\_NDISC\_NEEDED).

**fptun-exception-ether-dst <uint64>** The number of FPTUN\_EXC\_ETHER\_DST exceptions (FPTUN\_EXC\_ETHER\_DST).

**fptun-exception-fpc <uint64>** The number of FPTUN\_EXC\_FPC exceptions (FPTUN\_EXC\_FPC).

**fptun-exception-icmp-needed <uint64>** The number of FPTUN\_EXC\_ICMP\_NEEDED exceptions (FPTUN\_EXC\_ICMP\_NEEDED).

**fptun-exception-ike-needed <uint64>** The number of FPTUN\_EXC\_IKE\_NEEDED exceptions (FPTUN\_EXC\_IKE\_NEEDED).

**fptun-exception-ip-dst <uint64>** The number of FPTUN\_EXC\_IP\_DST exceptions (FPTUN\_EXC\_IP\_DST).

**fptun-exception-ip-pmtu <uint64>** The number of FPTUN\_EXC\_IP\_PMTU exceptions (FPTUN\_EXC\_IP\_PMTU).

**fptun-exception-ndisc-needed <uint64>** The number of FPTUN\_EXC\_NDISC\_NEEDED exceptions (FPTUN\_EXC\_NDISC\_NEEDED).

**fptun-exception-nf-func <uint64>** The number of FPTUN\_EXC\_NF\_FUNC exceptions (FPTUN\_EXC\_NF\_FUNC).

**fptun-exception-replaywin <uint64>** The number of FPTUN\_EXC\_REPLAYWIN exceptions (FPTUN\_EXC\_REPLAYWIN).

**fptun-exception-socket <uint64>** The number of FPTUN\_EXC\_SOCKET exceptions (FPTUN\_EXC\_SOCKET).

**fptun-exception-sp-func <uint64>** The number of FPTUN\_EXC\_SP\_FUNC exceptions (FPTUN\_EXC\_SP\_FUNC).

**fptun-exception-tap <uint64>** The number of FPTUN\_EXC\_TAP exceptions (FPTUN\_EXC\_TAP).

**fptun-exception-undef <uint64>** The number of FPTUN\_EXC\_UNDEF exceptions (FPTUN\_EXC\_UNDEF).

**fptun-exception-vnb-to-vnb <uint64>** The number of FPTUN\_EXC\_VNB\_TO\_VNB exceptions (FPTUN\_EXC\_VNB\_TO\_VNB).

**local-exception-type** Local exception types statistics.

- fptun-basic-exception <uint64>** Number of basic exception packets for Linux reception processing (FPTUN\_BASIC\_EXCEPT).
- fptun-eth-input-exception <uint64>** Number of FPTUN exception packets for Linux ethernet input processing (FPTUN\_ETH\_INPUT\_EXCEPT).
- fptun-eth-novnb-input-exception <uint64>** Number of FPTUN exception packets for Linux processing skipping VNB (FPTUN\_ETH\_NOVNB\_INPUT\_EXCEPT).
- fptun-eth-sp-output-req <uint64>** Number of FPTUN exception packets from Linux for fast path ethernet processing (FPTUN\_ETH\_SP\_OUTPUT\_REQ).
- fptun-iface-input-exception <uint64>** Number of FPTUN exception packets for Linux VNB iface processing (FPTUN\_IFACE\_INPUT\_EXCEPT).
- fptun-ipsec-sp-output-req <uint64>** Number of FPTUN exception packets from Linux for fast path IPsec processing (FPTUN\_IPSEC\_SP\_OUTPUT\_REQ).
- fptun-ipv4-ipsecdone-input-exception <uint64>** Number of FPTUN exception packets for Linux IPv4 input after IPsec processing (FPTUN\_IPV4\_IPSECDONE\_INPUT\_EXCEPT).
- fptun-ipv4-ipsecdone-output-exception <uint64>** Number of FPTUN exception packets for Linux IPv4 output after IPsec processing (FPTUN\_IPV4\_IPSECDONE\_OUTPUT\_EXCEPT).
- fptun-ipv4-natdone-input-exception <uint64>** Number of FPTUN exception packets for Linux IPv4 input after NAT processing (FPTUN\_IPV4\_NATDONE\_INPUT\_EXCEPT).
- fptun-ipv4-output-exception <uint64>** Number of FPTUN exception packets for Linux IPv4 output (FPTUN\_IPV4\_OUTPUT\_EXCEPT).
- fptun-ipv4-sp-output-req <uint64>** Number of FPTUN exception packets from Linux for fast path IPv4 processing (FPTUN\_IPV4\_SP\_OUTPUT\_REQ).
- fptun-ipv6-ipsecdone-input-exception <uint64>** Number of FPTUN exception packets for Linux IPv6 input after IPsec processing (FPTUN\_IPV6\_IPSECDONE\_INPUT\_EXCEPT).
- fptun-ipv6-ipsecdone-output-exception <uint64>** Number of FPTUN exception packets for Linux IPv6 output after IPsec processing (FPTUN\_IPV6\_IPSECDONE\_OUTPUT\_EXCEPT).
- fptun-ipv6-output-exception <uint64>** Number of FPTUN exception packets for Linux IPv6 output (FPTUN\_IPV6\_OUTPUT\_EXCEPT).
- fptun-ipv6-sp-output-req <uint64>** Number of FPTUN exception packets from Linux for fast path IPv6 processing (FPTUN\_IPV6\_SP\_OUTPUT\_REQ).
- fptun-output-exception <uint64>** Number of FPTUN exception packets for Linux interface output processing (FPTUN\_OUTPUT\_EXCEPT).
- fptun-rfps-update <uint64>** Number of FPTUN exception packets for fast path statistics processing (FPTUN\_RFPS\_UPDATE).
- fptun-tap <uint64>** Number of FPTUN exception packets for Linux TAP (tcpdump) (FPTUN\_TAP).

**fptun-traffic-generator-msg <uint64>** Number of FPTUN exception packets for fast path traffic generator processing (FPTUN\_TRAFFIC\_GEN\_MSG).

**fptun-vnb2vnb-fp-to-linux-exception <uint64>** Number of FPTUN VNB exception packets for Linux VNB input processing (FPTUN\_VNB2VNB\_FP\_TO\_LINUX\_EXCEPT).

**fptun-vnb2vnb-linux-to-fp-exception <uint64>** Number of FPTUN exception packets for Linux VNB processing (FPTUN\_VNB2VNB\_LINUX\_TO\_FP\_EXCEPT).

**local-fptun-exceptions <uint64>** The number of local fptun exceptions (LocalFPTunExceptions).

**qos-sched** QoS scheduler service statistics.

**interface** Interface QoS statistics.

**name <string>** Interface name.

**index <uint64>** Interface index.

**vrfid <uint64>** Vrf Id.

**enqueue-drop-no-class-packets <uint64>** The number of packets that were dropped because they matched no class (enq\_drop\_noclass\_pkts).

**enqueue-drop-policer-packets <uint64>** The number of packets that were dropped by a policer (enq\_drop\_meter\_pkts).

**enqueue-drop-queue-full-packets <uint64>** The number of packets that were dropped because a queue was full (enq\_drop\_qfull\_pkts).

**enqueue-success-packets <uint64>** The number of packets that were enqueued (enq\_ok\_pkts).

**transmit-drop-packets <uint64>** The number of packets that were dropped during transmission (xmit\_drop\_pkts).

**transmit-success-packets <uint64>** The number of packets that were transmitted (xmit\_ok\_pkts).

**class** Fast-path class statistics.

**class-id <uint64>** Class Id.

**index <uint64>** Class index.

**enqueue-drop-policer-packets <uint64>** The number of packets that were dropped by a policer (enq\_drop\_meter\_pkts).

**enqueue-drop-queue-full-packets <uint64>** The number of packets that were dropped because a queue was full (enq\_drop\_qfull\_pkts).

**enqueue-success-packets <uint64>** The number of packets that were enqueued (enq\_ok\_pkts).

**transmit-drop-packets <uint64>** The number of packets that were dropped during transmission (xmit\_drop\_pkts).

**transmit-success-packets** <uint64> The number of packets that were transmitted (xmit\_ok\_pkts).

**qos-rate-limit** QoS rate limit service statistics.

**green** Green statistics.

**packets** <uint64> Number of green packets.

**bytes** <uint64> Number of green bytes.

**yellow** Yellow statistics.

**packets** <uint64> Number of yellow packets.

**bytes** <uint64> Number of yellow bytes.

**red** Red statistics.

**packets** <uint64> Number of red packets.

**bytes** <uint64> Number of red bytes.

## show ike

---

**Note:** requires a Turbo IPsec Application License.

---

```
vrouter> show ike [vrf VRF] counters [vpn <string>] ike-sa [details] [vpn <string>] \
...          [remote-ip <string>] [remote-id <string>] [state STATE] ike-sa-count \
...          [state STATE] ipsec-sa-count [fastpath]
```

Show filtered SA state or general information.

## Input Parameters

**vrf VRF** Show objects in selected netns only.

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

**counters** [vpn <string>] Show IKE counters.

**vpn <string>** Show counters for selected VPN.

**ike-sa** [details] [vpn <string>] [remote-ip <string>] [remote-id <string>] [state STATE]  
Show SA state.

**details** Show detailed output.

**vpn <string>** Show SA for selected VPN.

**remote-ip <string>** Show SAs to selected remote-ip.

**remote-id <string>** Show SAs to selected remote-id.

**state STATE** Show SAs in selected state.

| STATE values | Description  |
|--------------|--|
| created      | IKE SA just got created, but is not initiating nor responding yet. |
| connecting   | IKE SA gets initiated actively or passively.                       |
| established  | IKE SA is fully established.                                       |
| passive      | IKE SA is managed externally and does not process messages.        |
| rekeying     | IKE SA rekeying in progress.                                       |
| rekeyed      | IKE SA has been rekeyed (or is redundant).                         |
| deleting     | IKE SA deletion in progress.                                       |
| destroying   | IKE SA object gets destroyed.                                      |

**ike-sa-count [state STATE]** Show SA count.

**state STATE** Only count SAs in selected state.

| STATE values | Description  |
|--------------|--|
| created      | IKE SA just got created, but is not initiating nor responding yet. |
| connecting   | IKE SA gets initiated actively or passively.                       |
| established  | IKE SA is fully established.                                       |
| passive      | IKE SA is managed externally and does not process messages.        |
| rekeying     | IKE SA rekeying in progress.                                       |
| rekeyed      | IKE SA has been rekeyed (or is redundant).                         |
| deleting     | IKE SA deletion in progress.                                       |
| destroying   | IKE SA object gets destroyed.                                      |

**ipsec-sa-count [fastpath]** Show IPsec SA count (default is from Linux).

**fastpath** Show IPsec SA count from Fast-Path.

### show cg-nat pool-usage

---

**Note:** requires a Turbo CG-NAT Application License.

---

```
vrouter> show cg-nat pool-usage [vrf <string>] pool-name <string> [address ADDRESS]
```

Show address usage of a CG-NAT pool.

## Input Parameters

**vrf** <string> VRF.

**pool-name** <string> (mandatory) IP address pool name.

**address** ADDRESS IP address in the pool.

|         |                  |
|---------|------------------|
| ADDRESS | An IPv4 address. |
|---------|------------------|

## show cg-nat port-usage

---

**Note:** requires a Turbo CG-NAT Application License.

---

```
vrouter> show cg-nat port-usage [vrf <string>] rule-id <uint16> user-address USER-ADDRESS
```

Show port usage of a CG-NAT user.

## Input Parameters

**vrf** <string> VRF.

**rule-id** <uint16> (mandatory) Rule id.

**user-address** USER-ADDRESS (mandatory) User IP address.

| USER-ADDRESS values | Description      |
|---------------------|------------------|
| <ip-address>        | An IPv4 address. |
| <ip-address>        | An IPv6 address. |

## show cg-nat user

---

**Note:** requires a Turbo CG-NAT Application License.

---

```
vrouter> show cg-nat user [vrf <string>] rule-id <uint16> [user-address USER-ADDRESS] \
↳ [threshold-errors <uint32>] \
... [usage-min <uint8>]
```

Show user(s) of a CG-NAT rule.

## Input Parameters

**vrf** <string> VRF.

**rule-id** <uint16> (mandatory) Rule id.

**user-address** USER-ADDRESS User IP address.

| USER-ADDRESS values | Description      |
|---------------------|------------------|
| <ip-address>        | An IPv4 address. |
| <ip-address>        | An IPv6 address. |

**threshold-errors** <uint32> Users having more errors than a given threshold.

**usage-min** <uint8> Users usage by at least the given rate.

## show cg-nat blocks

---

**Note:** requires a Turbo CG-NAT Application License.

---

```
vrouter> show cg-nat blocks [vrf <string>] rule-id <uint16> user-address USER-ADDRESS
```

Show blocks of a CG-NAT user.

## Input Parameters

**vrf** <string> VRF.

**rule-id** <uint16> (mandatory) Rule id.

**user-address** USER-ADDRESS (mandatory) User IP address.

| USER-ADDRESS values | Description      |
|---------------------|------------------|
| <ip-address>        | An IPv4 address. |
| <ip-address>        | An IPv6 address. |

### show cg-nat contracks

**Note:** requires a Turbo CG-NAT Application License.

```
vrouters> show cg-nat contracks [vrf <string>] rule-id <uint16> user-address USER-
↳ADDRESS forward \
...      [peer-address PEER-ADDRESS] backward [peer-address PEER-ADDRESS] \
...      protocol tcp [peer-port PEER-PORT] udp [peer-port PEER-PORT] \
...      icmp [peer-id <uint16>] icmpv6 [peer-id <uint16>] gre-pptp [key <uint16>
↳]
```

Show contracks of a CG-NAT user.

### Input Parameters

**vrf <string>** VRF.

**rule-id <uint16> (mandatory)** Rule id.

**user-address USER-ADDRESS (mandatory)** User IP address.

| USER-ADDRESS values | Description      |
|---------------------|------------------|
| <ip-address>        | An IPv4 address. |
| <ip-address>        | An IPv6 address. |

**forward [peer-address PEER-ADDRESS]** Filter by IP and/or port using the forward tuple (the default).

**peer-address PEER-ADDRESS** Forward peer IPv4/IPv6 address.

| PEER-ADDRESS values | Description      |
|---------------------|------------------|
| <A.B.C.D>           | An IPv4 address. |
| <X:X::X:X>          | An IPv6 address. |

**backward [peer-address PEER-ADDRESS]** Filter by IP and/or port using the backward tuple.

**peer-address PEER-ADDRESS** Backward peer IPv4 address.

|              |                  |
|--------------|------------------|
| PEER-ADDRESS | An IPv4 address. |
|--------------|------------------|

**protocol tcp [peer-port PEER-PORT] udp [peer-port PEER-PORT] icmp [peer-id <uint16>] icmpv6 [peer-id <uint16>]**  
Filter contracks per protocol.

**tcp [peer-port PEER-PORT]** Show only contracks using the TCP protocol.

**peer-port PEER-PORT** Peer port.

|           |   |
|-----------|---|
| PEER-PORT | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-----------|---|

**udp [peer-port PEER-PORT]** Show only conntracks using the UDP protocol.

**peer-port PEER-PORT** Peer port.

|           |   |
|-----------|---|
| PEER-PORT | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-----------|---|

**icmp [peer-id <uint16>]** Show only conntracks using the ICMP protocol.

**peer-id <uint16>** ICMP peer identifier.

**icmpv6 [peer-id <uint16>]** Show only conntracks using the ICMPv6 protocol.

**peer-id <uint16>** ICMPv6 peer identifier.

**gre-pptp [key <uint16>]** Show only conntracks using the GRE-PPTP protocol.

**key <uint16>** GRE key.

### show cg-nat conntrack-statistics

---

**Note:** requires a Turbo CG-NAT Application License.

---

```
vrouter> show cg-nat conntrack-statistics [vrf <string>] rule-id <uint16>
```

Show conntracks usage statistics of a CG-NAT rule.

#### Input Parameters

**vrf <string>** VRF.

**rule-id <uint16> (mandatory)** Rule id.

### show cg-nat port-statistics

---

**Note:** requires a Turbo CG-NAT Application License.

---

```
vrouter> show cg-nat port-statistics [vrf <string>] rule-id <uint16> [protocol_
↳PROTOCOL]
```

Show port usage statistics of a CG-NAT rule.

### Input Parameters

**vrf** <string> VRF.

**rule-id** <uint16> (mandatory) Rule id.

**protocol** PROTOCOL Protocol.

| PROTOCOL values | Description  |
|-----------------|--|
| tcp             | Transmission Control Protocol.                                       |
| udp             | User Datagram Protocol.  |
| icmp            | Internet Control Message Protocol.                                   |
| gre-pptp        | Generic Routing Encapsulation for Point-to-Point Tunneling Protocol. |

### show cg-nat block-statistics

---

**Note:** requires a Turbo CG-NAT Application License.

---

```
vrouter> show cg-nat block-statistics [vrf <string>] rule-id <uint16> [protocol_
↳PROTOCOL]
```

Show block usage statistics of a CG-NAT rule.

### Input Parameters

**vrf** <string> VRF.

**rule-id** <uint16> (mandatory) Rule id.

**protocol** PROTOCOL Protocol.

| PROTOCOL values | Description  |
|-----------------|--|
| tcp             | Transmission Control Protocol.                                       |
| udp             | User Datagram Protocol.  |
| icmp            | Internet Control Message Protocol.                                   |
| gre-pptp        | Generic Routing Encapsulation for Point-to-Point Tunneling Protocol. |

### show cg-nat hash-table-statistics

---

**Note:** requires a Turbo CG-NAT Application License.

---

```
vrouters> show cg-nat hash-table-statistics
```

Show hash table statistics.

### show cg-nat mempool-usage

---

**Note:** requires a Turbo CG-NAT Application License.

---

```
vrouters> show cg-nat mempool-usage
```

Show memory pool usage.

### show cg-nat statistics

---

**Note:** requires a Turbo CG-NAT Application License.

---

```
vrouters> show cg-nat statistics
```

Show global statistics.

### show ha-neighbor

---

**Note:** requires a Turbo Router Network License.

---

```
vrouters> show ha-neighbor [vrf VRF] state
```

Show high-availability neighbor state.

## Input Parameters

**vrf VRF** Specify the VRF.

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

**state (mandatory)** Show high-availability neighbor state.

## show ha-conntrack

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> show ha-conntrack [vrf VRF] [state] [cache CACHE]
```

Show high-availability conntrack state.

## Input Parameters

**vrf VRF** Specify the VRF.

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

**state** Show high-availability conntrack state.

**cache CACHE** Display cache content.

| CACHE values | Description                            |
|--------------|--|
| internal     | Display content of the internal cache. |
| external     | Display content of the external cache. |

### 3.2.3 flush

#### flush bgp

**Note:** requires a Turbo Router Network License.

```

vrouters> flush bgp [vrf <string>] ipv4 unicast [as AS] [all] [neighbor NEIGHBOR] \
...          [external] [neighbor-group <string>] [soft SOFT] multicast [as AS] \
...          [all] [neighbor NEIGHBOR] [external] [neighbor-group <string>] \
...          [soft SOFT] labeled-unicast [as AS] [all] [neighbor NEIGHBOR] \
...          [external] [neighbor-group <string>] [soft SOFT] flowspec [as AS] \
...          [all] [neighbor NEIGHBOR] [external] [neighbor-group <string>] \
...          [soft SOFT] vpn [as AS] [all] [neighbor NEIGHBOR] [external] \
...          [neighbor-group <string>] [soft SOFT] ipv6 unicast [as AS] [all] \
...          [neighbor NEIGHBOR] [external] [neighbor-group <string>] [soft SOFT] \
...          multicast [as AS] [all] [neighbor NEIGHBOR] [external] [neighbor-group
↪<string>] \
...          [soft SOFT] labeled-unicast [as AS] [all] [neighbor NEIGHBOR] \
...          [external] [neighbor-group <string>] [soft SOFT] flowspec [as AS] \
...          [all] [neighbor NEIGHBOR] [external] [neighbor-group <string>] \
...          [soft SOFT] vpn [as AS] [all] [neighbor NEIGHBOR] [external] \
...          [neighbor-group <string>] [soft SOFT]

```

Flush BGP information.

#### Input Parameters

**vrf <string>** Specify the VRF.

**ipv4 unicast [as AS] [all] [neighbor NEIGHBOR] [external] [neighbor-group <string>] [soft SOFT] m**  
Flush information about BGP IPv4.

**unicast [as AS] [all] [neighbor NEIGHBOR] [external] [neighbor-group <string>] [soft SOFT]**  
Flush information for unicast address family.

**as AS** Flush neighbors with the AS number.

|    |   |
|----|---|
| AS | A numeric identifier for an autonomous system (AS). An AS is a single domain, under common administrative control, which forms a unit of routing policy. Autonomous systems can be assigned a 2-byte identifier, or a 4-byte identifier which may have public or private scope. Private ASNs are assigned from dedicated ranges. Public ASNs are assigned from ranges allocated by IANA to the regional internet registries (RIRs). |
|----|---|

**all** Flush all neighbors.

**neighbor NEIGHBOR** BGP neighbor address to flush.

| NEIGHBOR values | Description      |
|-----------------|------------------|
| <A.B.C.D>       | An IPv4 address. |
| <X:X::X:X>      | An IPv6 address. |

**external** Flush all external neighbors.

**neighbor-group <string>** Flush all members of the neighbor group.

**soft SOFT** Soft reconfigure inbound and/or outbound updates.

| SOFT values | Description   |
|-------------|---|
| in          | Send route-refresh unless using ‘soft-reconfiguration inbound’. |
| out         | Resend all outbound updates.                                    |
| both        | Soft reconfigure inbound and outbound updates.                  |

**multicast [as AS] [all] [neighbor NEIGHBOR] [external] [neighbor-group <string>] [soft SOFT]**  
 Flush information for multicast address family.

**as AS** Flush neighbors with the AS number.

|    |   |
|----|---|
| AS | A numeric identifier for an autonomous system (AS). An AS is a single domain, under common administrative control, which forms a unit of routing policy. Autonomous systems can be assigned a 2-byte identifier, or a 4-byte identifier which may have public or private scope. Private ASNs are assigned from dedicated ranges. Public ASNs are assigned from ranges allocated by IANA to the regional internet registries (RIRs). |
|----|---|

**all** Flush all neighbors.

**neighbor NEIGHBOR** BGP neighbor address to flush.

| NEIGHBOR values | Description      |
|-----------------|------------------|
| <A.B.C.D>       | An IPv4 address. |
| <X:X::X:X>      | An IPv6 address. |

**external** Flush all external neighbors.

**neighbor-group <string>** Flush all members of the neighbor group.

**soft SOFT** Soft reconfigure inbound and/or outbound updates.

| SOFT values | Description   |
|-------------|---|
| in          | Send route-refresh unless using ‘soft-reconfiguration inbound’. |
| out         | Resend all outbound updates.                                    |
| both        | Soft reconfigure inbound and outbound updates.                  |

**labeled-unicast** [**as** AS] [**all**] [**neighbor** NEIGHBOR] [**external**] [**neighbor-group** <string>] [**soft**

Flush information for labeled unicast address family.

**as** AS Flush neighbors with the AS number.

|    |   |
|----|---|
| AS | A numeric identifier for an autonomous system (AS). An AS is a single domain, under common administrative control, which forms a unit of routing policy. Autonomous systems can be assigned a 2-byte identifier, or a 4-byte identifier which may have public or private scope. Private ASNs are assigned from dedicated ranges. Public ASNs are assigned from ranges allocated by IANA to the regional internet registries (RIRs). |
|----|---|

**all** Flush all neighbors.

**neighbor** NEIGHBOR BGP neighbor address to flush.

| NEIGHBOR values | Description      |
|-----------------|------------------|
| <A.B.C.D>       | An IPv4 address. |
| <X:X::X:X>      | An IPv6 address. |

**external** Flush all external neighbors.

**neighbor-group** <string> Flush all members of the neighbor group.

**soft** SOFT Soft reconfigure inbound and/or outbound updates.

| SOFT values | Description   |
|-------------|---|
| in          | Send route-refresh unless using ‘soft-reconfiguration inbound’. |
| out         | Resend all outbound updates.                                    |
| both        | Soft reconfigure inbound and outbound updates.                  |

**flowspec** [**as** AS] [**all**] [**neighbor** NEIGHBOR] [**external**] [**neighbor-group** <string>] [**soft** SOFT]

Flush information for flowspec address family.

**as** AS Flush neighbors with the AS number.

|    |   |
|----|---|
| AS | A numeric identifier for an autonomous system (AS). An AS is a single domain, under common administrative control, which forms a unit of routing policy. Autonomous systems can be assigned a 2-byte identifier, or a 4-byte identifier which may have public or private scope. Private ASNs are assigned from dedicated ranges. Public ASNs are assigned from ranges allocated by IANA to the regional internet registries (RIRs). |
|----|---|

**all** Flush all neighbors.

**neighbor** NEIGHBOR BGP neighbor address to flush.

| NEIGHBOR values | Description      |
|-----------------|------------------|
| <A.B.C.D>       | An IPv4 address. |
| <X:X::X:X>      | An IPv6 address. |

**external** Flush all external neighbors.

**neighbor-group <string>** Flush all members of the neighbor group.

**soft SOFT** Soft reconfigure inbound and/or outbound updates.

| SOFT values | Description   |
|-------------|---|
| in          | Send route-refresh unless using ‘soft-reconfiguration inbound’. |
| out         | Resend all outbound updates.                                    |
| both        | Soft reconfigure inbound and outbound updates.                  |

**vpn [as AS] [all] [neighbor NEIGHBOR] [external] [neighbor-group <string>] [soft SOFT]**  
 Flush information for VPN address family.

**as AS** Flush neighbors with the AS number.

|           |   |
|-----------|---|
| <b>AS</b> | A numeric identifier for an autonomous system (AS). An AS is a single domain, under common administrative control, which forms a unit of routing policy. Autonomous systems can be assigned a 2-byte identifier, or a 4-byte identifier which may have public or private scope. Private ASNs are assigned from dedicated ranges. Public ASNs are assigned from ranges allocated by IANA to the regional internet registries (RIRs). |
|-----------|---|

**all** Flush all neighbors.

**neighbor NEIGHBOR** BGP neighbor address to flush.

| NEIGHBOR values | Description      |
|-----------------|------------------|
| <A.B.C.D>       | An IPv4 address. |
| <X:X::X:X>      | An IPv6 address. |

**external** Flush all external neighbors.

**neighbor-group <string>** Flush all members of the neighbor group.

**soft SOFT** Soft reconfigure inbound and/or outbound updates.

| SOFT values | Description   |
|-------------|---|
| in          | Send route-refresh unless using ‘soft-reconfiguration inbound’. |
| out         | Resend all outbound updates.                                    |
| both        | Soft reconfigure inbound and outbound updates.                  |

**ipv6 unicast** [**as AS**] [**all**] [**neighbor NEIGHBOR**] [**external**] [**neighbor-group <string>**] [**soft SOFT**] m  
 Flush information about BGP IPv6.

**unicast** [**as AS**] [**all**] [**neighbor NEIGHBOR**] [**external**] [**neighbor-group <string>**] [**soft SOFT**]  
 Flush information for unicast address family.

**as AS** Flush neighbors with the AS number.

|    |   |
|----|---|
| AS | A numeric identifier for an autonomous system (AS). An AS is a single domain, under common administrative control, which forms a unit of routing policy. Autonomous systems can be assigned a 2-byte identifier, or a 4-byte identifier which may have public or private scope. Private ASNs are assigned from dedicated ranges. Public ASNs are assigned from ranges allocated by IANA to the regional internet registries (RIRs). |
|----|---|

**all** Flush all neighbors.

**neighbor NEIGHBOR** BGP neighbor address to flush.

| NEIGHBOR values | Description      |
|-----------------|------------------|
| <A.B.C.D>       | An IPv4 address. |
| <X:X::X:X>      | An IPv6 address. |

**external** Flush all external neighbors.

**neighbor-group <string>** Flush all members of the neighbor group.

**soft SOFT** Soft reconfigure inbound and/or outbound updates.

| SOFT values | Description   |
|-------------|---|
| in          | Send route-refresh unless using ‘soft-reconfiguration inbound’. |
| out         | Resend all outbound updates.                                    |
| both        | Soft reconfigure inbound and outbound updates.                  |

**multicast** [**as AS**] [**all**] [**neighbor NEIGHBOR**] [**external**] [**neighbor-group <string>**] [**soft SOFT**]  
 Flush information for multicast address family.

**as AS** Flush neighbors with the AS number.

|    |   |
|----|---|
| AS | A numeric identifier for an autonomous system (AS). An AS is a single domain, under common administrative control, which forms a unit of routing policy. Autonomous systems can be assigned a 2-byte identifier, or a 4-byte identifier which may have public or private scope. Private ASNs are assigned from dedicated ranges. Public ASNs are assigned from ranges allocated by IANA to the regional internet registries (RIRs). |
|----|---|

**all** Flush all neighbors.

**neighbor NEIGHBOR** BGP neighbor address to flush.

| NEIGHBOR values | Description      |
|-----------------|------------------|
| <A.B.C.D>       | An IPv4 address. |
| <X:X::X:X>      | An IPv6 address. |

**external** Flush all external neighbors.

**neighbor-group <string>** Flush all members of the neighbor group.

**soft SOFT** Soft reconfigure inbound and/or outbound updates.

| SOFT values | Description   |
|-------------|---|
| in          | Send route-refresh unless using ‘soft-reconfiguration inbound’. |
| out         | Resend all outbound updates.                                    |
| both        | Soft reconfigure inbound and outbound updates.                  |

**labeled-unicast [as AS] [all] [neighbor NEIGHBOR] [external] [neighbor-group <string>] [soft**  
 Flush information for labeled unicast address family.

**as AS** Flush neighbors with the AS number.

|    |   |
|----|---|
| AS | A numeric identifier for an autonomous system (AS). An AS is a single domain, under common administrative control, which forms a unit of routing policy. Autonomous systems can be assigned a 2-byte identifier, or a 4-byte identifier which may have public or private scope. Private ASNs are assigned from dedicated ranges. Public ASNs are assigned from ranges allocated by IANA to the regional internet registries (RIRs). |
|----|---|

**all** Flush all neighbors.

**neighbor NEIGHBOR** BGP neighbor address to flush.

| NEIGHBOR values | Description      |
|-----------------|------------------|
| <A.B.C.D>       | An IPv4 address. |
| <X:X::X:X>      | An IPv6 address. |

**external** Flush all external neighbors.

**neighbor-group <string>** Flush all members of the neighbor group.

**soft SOFT** Soft reconfigure inbound and/or outbound updates.

| SOFT values | Description   |
|-------------|---|
| in          | Send route-refresh unless using ‘soft-reconfiguration inbound’. |
| out         | Resend all outbound updates.                                    |
| both        | Soft reconfigure inbound and outbound updates.                  |

**flowspec [as AS] [all] [neighbor NEIGHBOR] [external] [neighbor-group <string>] [soft SOFT]**  
 Flush information for flowspec address family.

**as AS** Flush neighbors with the AS number.

|    |   |
|----|---|
| AS | A numeric identifier for an autonomous system (AS). An AS is a single domain, under common administrative control, which forms a unit of routing policy. Autonomous systems can be assigned a 2-byte identifier, or a 4-byte identifier which may have public or private scope. Private ASNs are assigned from dedicated ranges. Public ASNs are assigned from ranges allocated by IANA to the regional internet registries (RIRs). |
|----|---|

**all** Flush all neighbors.

**neighbor NEIGHBOR** BGP neighbor address to flush.

| NEIGHBOR values | Description      |
|-----------------|------------------|
| <A.B.C.D>       | An IPv4 address. |
| <X:X::X:X>      | An IPv6 address. |

**external** Flush all external neighbors.

**neighbor-group <string>** Flush all members of the neighbor group.

**soft SOFT** Soft reconfigure inbound and/or outbound updates.

| SOFT values | Description   |
|-------------|---|
| in          | Send route-refresh unless using ‘soft-reconfiguration inbound’. |
| out         | Resend all outbound updates.                                    |
| both        | Soft reconfigure inbound and outbound updates.                  |

**vpn [as AS] [all] [neighbor NEIGHBOR] [external] [neighbor-group <string>] [soft SOFT]**  
 Flush information for VPN address family.

**as AS** Flush neighbors with the AS number.

|    |   |
|----|---|
| AS | A numeric identifier for an autonomous system (AS). An AS is a single domain, under common administrative control, which forms a unit of routing policy. Autonomous systems can be assigned a 2-byte identifier, or a 4-byte identifier which may have public or private scope. Private ASNs are assigned from dedicated ranges. Public ASNs are assigned from ranges allocated by IANA to the regional internet registries (RIRs). |
|----|---|

**all** Flush all neighbors.

**neighbor NEIGHBOR** BGP neighbor address to flush.

| NEIGHBOR values | Description      |
|-----------------|------------------|
| <A.B.C.D>       | An IPv4 address. |
| <X:X::X:X>      | An IPv6 address. |

**external** Flush all external neighbors.

**neighbor-group <string>** Flush all members of the neighbor group.

**soft SOFT** Soft reconfigure inbound and/or outbound updates.

| SOFT values | Description   |
|-------------|---|
| in          | Send route-refresh unless using ‘soft-reconfiguration inbound’. |
| out         | Resend all outbound updates.                                    |
| both        | Soft reconfigure inbound and outbound updates.                  |

## flush ospf

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> flush ospf [vrf <string>] interface <ifname>
```

Flush OSPF information.

### Input Parameters

**vrf <string>** Specify the VRF.

**interface <ifname> (mandatory)** The name of the network interface to be cleared.

## flush ospf6

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> flush ospf6 interface <ifname>
```

Flush OSPFv3 information.

## Input Parameters

**interface** <i fname> (mandatory) The name of the network interface to be cleared.

## flush nhrp

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> flush nhrp [vrf <string>] [cache] [shortcut]
```

Flush NHRP information.

## Input Parameters

**vrf** <string> Specify the VRF.

**cache** NHRP dynamic cache entries.

**shortcut** NHRP shortcut entries.

## flush nhrp6

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> flush nhrp6 [vrf <string>] [cache] [shortcut]
```

Flush NHRP information.

## Input Parameters

**vrf** <string> Specify the VRF.

**cache** NHRP dynamic cache entries.

**shortcut** NHRP shortcut entries.

## flush ike ike-sa

---

**Note:** requires a Turbo IPsec Application License.

---

```
vrouter> flush ike ike-sa [vrf VRF] [vpn <string>] [unique-id <string>]
```

Flush IKE SA.

### Input Parameters

**vrf VRF** Flush objects in selected netns only.

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

**vpn <string>** Flush SA for selected VPN.

**unique-id <string>** Flush SA with this unique id.

## flush ike child-sa

---

**Note:** requires a Turbo IPsec Application License.

---

```
vrouter> flush ike child-sa [vrf VRF] [name <string>] [unique-id <string>]
```

Flush child SA.

### Input Parameters

**vrf VRF** Flush objects in selected netns only.

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

**name <string>** Flush SA with this name.

**unique-id <string>** Flush SA with this unique id.

## flush cg-nat statistics

---

**Note:** requires a Turbo CG-NAT Application License.

---

```
vrouter> flush cg-nat statistics
```

Reset statistics.

## flush cg-nat user

---

**Note:** requires a Turbo CG-NAT Application License.

---

```
vrouter> flush cg-nat user [vrf <string>] rule-id <uint16> [user-address USER-ADDRESS]
```

Flush one of all users of a CG-NAT rule.

## Input Parameters

**vrf <string>** VRF.

**rule-id <uint16> (mandatory)** Rule id.

**user-address USER-ADDRESS** User IP address.

| USER-ADDRESS values | Description      |
|---------------------|------------------|
| <ip-address>        | An IPv4 address. |
| <ip-address>        | An IPv6 address. |

## flush cg-nat user-statistics

---

**Note:** requires a Turbo CG-NAT Application License.

---

```
vrouter> flush cg-nat user-statistics [vrf <string>] rule-id <uint16> [user-address_
↳USER-ADDRESS] [threshold-errors <uint32>] \
... [usage-min <uint8>]
```

Flush statistics of one or all users of a CG-NAT rule.

## Input Parameters

**vrf** <string> VRF.

**rule-id** <uint16> (mandatory) Rule id.

**user-address** USER-ADDRESS User IP address.

| USER-ADDRESS values | Description      |
|---------------------|------------------|
| <ip-address>        | An IPv4 address. |
| <ip-address>        | An IPv6 address. |

**threshold-errors** <uint32> Users having more errors than a given threshold.

**usage-min** <uint8> Users usage by at least the given rate.

## flush fast-path statistics

---

**Note:** requires a Turbo Router Network License.

---

```
vrouter> flush fast-path statistics
```

Flush fast-path statistics.

## 3.2.4 system

Global system configuration.

```
vrouter running config# system
```

### hostname

The hostname of the device – should be a single domain label, without the domain.

```
vrouter running config# system
vrouter running system# hostname HOSTNAME
```

**HOSTNAME** The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names **MUST** be encoded in punycode as described in RFC 3492.

**cp-mask**

**Note:** requires a Turbo Router Network License.

Cores on which control plane applications run.

```
vrouter running config# system
vrouter running system# cp-mask CP-MASK
```

| CP-MASK values | Description   |
|----------------|---|
| default        | Use all cores except fast path ones for control plane.                  |
| <cores-list>   | A comma-separated list of cores or core ranges. Example: '1,4-7,10-12'. |

**Default value**

default

**timezone**

The timezone of the device.

```
vrouter running config# system
vrouter running system# timezone TIMEZONE
```

| TIMEZONE values |
|-----------------|
| UTC             |

|                                |
|--------------------------------|
| TIMEZONE values                |
| GMT                            |
| Antarctica/McMurdo             |
| Antarctica/South_Pole          |
| Antarctica/Rothera             |
| Antarctica/Palmer              |
| Antarctica/Mawson              |
| Antarctica/Davis               |
| Antarctica/Casey               |
| Antarctica/Vostok              |
| Antarctica/DumontDUrville      |
| Antarctica/Syowa               |
| Antarctica/Macquarie           |
| America/Argentina/Buenos_Aires |
| America/Argentina/Cordoba      |
| America/Argentina/Salta        |
| America/Argentina/Jujuy        |
| America/Argentina/Tucuman      |
| America/Argentina/Catamarca    |
| America/Argentina/La_Rioja     |
| America/Argentina/San_Juan     |
| America/Argentina/Mendoza      |
| America/Argentina/San_Luis     |
| America/Argentina/Rio_Gallegos |
| America/Argentina/Ushuaia      |
| Australia/Lord_Howe            |
| Australia/Hobart               |
| Australia/Currie               |
| Australia/Melbourne            |
| Australia/Sydney               |
| Australia/Broken_Hill          |
| Australia/Brisbane             |
| Australia/Lindeman             |
| Australia/Adelaide             |
| Australia/Darwin               |
| Australia/Perth                |
| Australia/Eucla                |
| America/Noronha                |
| America/Belem                  |
| America/Fortaleza              |
| America/Recife                 |
| America/Araguaina              |

|                       |
|-----------------------|
| TIMEZONE values       |
| America/Maceio        |
| America/Bahia         |
| America/Sao_Paulo     |
| America/Campo_Grande  |
| America/Cuiaba        |
| America/Santarem      |
| America/Porto_Velho   |
| America/Boa_Vista     |
| America/Manaus        |
| America/Eirunepe      |
| America/Rio_Branco    |
| America/St_Johns      |
| America/Halifax       |
| America/Glace_Bay     |
| America/Moncton       |
| America/Goose_Bay     |
| America/Blanc-Sablon  |
| America/Montreal      |
| America/Toronto       |
| America/Nipigon       |
| America/Thunder_Bay   |
| America/Iqaluit       |
| America/Pangnirtung   |
| America/Resolute      |
| America/Atikokan      |
| America/Rankin_Inlet  |
| America/Winnipeg      |
| America/Rainy_River   |
| America/Regina        |
| America/Swift_Current |
| America/Edmonton      |
| America/Cambridge_Bay |
| America/Yellowknife   |
| America/Inuvik        |
| America/Creston       |
| America/Dawson_Creek  |
| America/Vancouver     |
| America/Whitehorse    |
| America/Dawson        |
| Africa/Kinshasa       |
| Africa/Lubumbashi     |

|                      |
|----------------------|
| TIMEZONE values      |
| America/Santiago     |
| Pacific/Easter       |
| Asia/Shanghai        |
| Asia/Harbin          |
| Asia/Chongqing       |
| Asia/Urumqi          |
| Asia/Kashgar         |
| America/Guayaquil    |
| Pacific/Galapagos    |
| Europe/Madrid        |
| Africa/Ceuta         |
| Atlantic/Canary      |
| Pacific/Chuuk        |
| Pacific/Pohnpei      |
| Pacific/Kosrae       |
| America/Godthab      |
| America/Danmarkshavn |
| America/Scoresbysund |
| America/Thule        |
| Asia/Jakarta         |
| Asia/Pontianak       |
| Asia/Makassar        |
| Asia/Jayapura        |
| Pacific/Tarawa       |
| Pacific/Enderbury    |
| Pacific/Kiritimati   |
| Asia/Almaty          |
| Asia/Qyzylorda       |
| Asia/Aqtobe          |
| Asia/Aqtau           |
| Asia/Oral            |
| Pacific/Majuro       |
| Pacific/Kwajalein    |
| Asia/Ulaanbaatar     |
| Asia/Hovd            |
| Asia/Choibalsan      |
| America/Mexico_City  |
| America/Cancun       |
| America/Merida       |
| America/Monterrey    |
| America/Matamoros    |

|                        |
|------------------------|
| TIMEZONE values        |
| America/Mazatlan       |
| America/Chihuahua      |
| America/Ojinaga        |
| America/Hermosillo     |
| America/Tijuana        |
| America/Santa_Isabel   |
| America/Bahia_Banderas |
| Asia/Kuala_Lumpur      |
| Asia/Kuching           |
| Pacific/Auckland       |
| Pacific/Chatham        |
| Pacific/Tahiti         |
| Pacific/Marquesas      |
| Pacific/Gambier        |
| Asia/Gaza              |
| Asia/Hebron            |
| Europe/Lisbon          |
| Atlantic/Madeira       |
| Atlantic/Azores        |
| Europe/Kaliningrad     |
| Europe/Moscow          |
| Europe/Volgograd       |
| Europe/Samara          |
| Asia/Yekaterinburg     |
| Asia/Omsk              |
| Asia/Novosibirsk       |
| Asia/Novokuznetsk      |
| Asia/Krasnoyarsk       |
| Asia/Irkutsk           |
| Asia/Yakutsk           |
| Asia/Vladivostok       |
| Asia/Sakhalin          |
| Asia/Magadan           |
| Asia/Kamchatka         |
| Asia/Anadyr            |
| Europe/Kiev            |
| Europe/Uzhgorod        |
| Europe/Zaporozhye      |
| Europe/Simferopol      |
| Pacific/Johnston       |
| Pacific/Midway         |

|  |
|--|
| TIMEZONE values  |
| Pacific/Wake   |
| America/New_York   |
| America/Detroit  |
| America/Kentucky/Louisville  |
| America/Kentucky/Monticello  |
| America/Indiana/Indianapolis   |
| America/Indiana/Vincennes  |
| America/Indiana/Winamac  |
| America/Indiana/Marengo  |
| America/Indiana/Petersburg   |
| America/Indiana/Vevay  |
| America/Chicago  |
| America/Indiana/Tell_City  |
| America/Indiana/Knox   |
| America/Menominee  |
| America/North_Dakota/Center  |
| America/North_Dakota/New_Salem   |
| America/North_Dakota/Beulah  |
| America/Denver   |
| America/Boise  |
| America/Shiprock   |
| America/Phoenix  |
| America/Los_Angeles  |
| America/Anchorage  |
| America/Juneau   |
| America/Sitka  |
| America/Yakutat  |
| America/Nome   |
| America/Adak   |
| America/Metlakatla   |
| Pacific/Honolulu   |
| Asia/Samarkand   |
| Asia/Tashkent  |
| Europe/Andorra Asia/Dubai Asia/Kabul America/Antigua America/Anguilla Europe/Tirane Asia/Yerevan Africa/Luanda Pacif |

**date (state only)**

The local time of the device.

```
vrouter> show state system date
```

**troubleshooting-report (state only)**

The existing troubleshooting reports available on the system.

```
vrouter> show state system troubleshooting-report
```

**traffic-capture (state only)**

The existing traffic captures available on the system.

```
vrouter> show state system traffic-capture
```

**network-stack**

---

**Note:** requires a Turbo Router Network License.

---

Network stack parameters.

```
vrouter running config# system network-stack
```

**bridge**

Bridge default parameters.

```
vrouter running config# system network-stack bridge
```

### call-ipv4-filtering

Call IPv4 filtering hooks on bridges.

```
vrouter running config# system network-stack bridge  
vrouter running bridge# call-ipv4-filtering true|false
```

#### Default value

false

### call-ipv6-filtering

Call IPv6 filtering hooks on bridges.

```
vrouter running config# system network-stack bridge  
vrouter running bridge# call-ipv6-filtering true|false
```

#### Default value

false

### icmp

ICMP default parameters.

```
vrouter running config# system network-stack icmp
```

### ignore-icmp-echo-broadcast

Ignore all ICMP ECHO and TIMESTAMP requests sent via broadcast or multicast.

```
vrouter running config# system network-stack icmp  
vrouter running icmp# ignore-icmp-echo-broadcast true|false
```

#### Default value

false

## rate-limit-icmp

The minimum time space that separates the sending of two consecutive ICMP packets. By default, such space is 1000 ms.

```
vrouter running config# system network-stack icmp
vrouter running icmp# rate-limit-icmp <uint16>
```

### Default value

1000

## rate-mask-icmp

Mask made of ICMP types for which rates are being limited.

```
vrouter running config# system network-stack icmp
vrouter running icmp# rate-mask-icmp RATE-MASK-ICMP
```

| RATE-MASK-ICMP values   | Description              |
|-------------------------|--------------------------|
| echo-reply              | Echo Reply.              |
| destination-unreachable | Destination Unreachable. |
| source-quench           | Source Quench.           |
| redirect                | Redirect.                |
| echo-request            | Echo Request.            |
| time-exceeded           | Time Exceeded.           |
| parameter-problem       | Parameter Problem.       |
| timestamp-request       | Timestamp Request.       |
| timestamp-reply         | Timestamp Reply.         |
| info-request            | Info Request.            |
| info-reply              | Info Reply.              |
| address-mask-request    | Address Mask Request.    |
| address-mask-reply      | Address Mask Reply.      |

### Default value

destination-unreachable source-quench time-exceeded parameter-problem

## ipv4

IPv4 default parameters.

```
vrouter running config# system network-stack ipv4
```

## forwarding

Enable IP forwarding.

```
vrouter running config# system network-stack ipv4  
vrouter running ipv4# forwarding true|false
```

### Default value

true

## send-redirects

Send ICMP redirect if host is on the same network than gateway.

```
vrouter running config# system network-stack ipv4  
vrouter running ipv4# send-redirects true|false
```

### Default value

true

## accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router.

```
vrouter running config# system network-stack ipv4  
vrouter running ipv4# accept-redirects true|false
```

### Default value

false

### accept-source-route

Accept packets with source route option.

```
vrouter running config# system network-stack ipv4
vrouter running ipv4# accept-source-route true|false
```

#### Default value

false

### arp-announce

Define different restriction levels for announcing the local source IP address from IP packets in ARP requests sent on interface. Increasing the restriction level gives more chance for receiving answer from the resolved target while decreasing the level announces more valid sender’s information.

```
vrouter running config# system network-stack ipv4
vrouter running ipv4# arp-announce ARP-ANNOUNCE
```

| ARP-ANNOUNCE values | Description  |
|---------------------|--|
| any                 | Use any local address, configured on any interface.  |
| avoid-not-in-subnet | Try to avoid local addresses that are not in the target’s subnet for this interface. This mode is useful when target hosts reachable via this interface require the source IP address in ARP requests to be part of their logical network configured on the receiving interface. When we generate the request we will check all our subnets that include the target IP and will preserve the source address if it is from such subnet. If there is no such subnet we select source address according to the rules for level 2, ‘best-local’.   |
| best-local          | Always use the best local address for this target. In this mode we ignore the source address in the IP packet and try to select local address that we prefer for talks with the target host. Such local address is selected by looking for primary IP addresses on all our subnets on the outgoing interface that include the target IP address. If no suitable local address is found we select the first local address we have on the outgoing interface or on all other interfaces, with the hope we will receive reply for our request and even sometimes no matter the source IP address we announce. |

#### Default value

any

## arp-filter

Allows to have multiple network interfaces on the same subnet, and have the ARPs for each interface be answered based on whether or not the kernel would route a packet from the ARP'd IP out that interface (therefore you must use source based routing for this to work). In other words it allows control of which cards (usually 1) will respond to an arp request.

```
vrouter running config# system network-stack ipv4
vrouter running ipv4# arp-filter true|false
```

### Default value

false

## arp-ignore

Define different modes for sending replies in response to received ARP requests that resolve local target IP addresses.

```
vrouter running config# system network-stack ipv4
vrouter running ipv4# arp-ignore ARP-IGNORE
```

| ARP-IGNORE values          | Description  |
|----------------------------|--|
| any                        | Reply for any local target IP address, configured on any interface.  |
| check-interface            | Reply only if the target IP address is local address configured on the incoming interface.   |
| check-interface-and-subnet | Reply only if the target IP address is local address configured on the incoming interface and both with the sender's IP address are part from same subnet on this interface. |
| ignore-scope               | Do not reply for local addresses configured with scope host, only resolutions for global and link addresses are replied.   |
| ignore-all                 | Do not reply for all local addresses.  |

### Default value

any

## log-invalid-addresses

Log packets with impossible addresses.

```
vrouter running config# system network-stack ipv4
vrouter running ipv4# log-invalid-addresses true|false
```

### Default value

false

## ipv6

IPv6 default parameters.

```
vrouter running config# system network-stack ipv6
```

## forwarding

Enable IPv6 forwarding.

```
vrouter running config# system network-stack ipv6
vrouter running ipv6# forwarding true|false
```

### Default value

true

## autoconfiguration

Autoconfigure addresses using Prefix Information in Router Advertisements.

```
vrouter running config# system network-stack ipv6
vrouter running ipv6# autoconfiguration true|false
```

### Default value

true

## accept-router-advert

Accept Router Advertisements.

```
vrouter running config# system network-stack ipv6
vrouter running ipv6# accept-router-advert ACCEPT-ROUTER-ADVERT
```

| ACCEPT-ROUTER-ADVERT values | Description   |
|-----------------------------|---|
| never                       | Do not accept Router Advertisements.                        |
| norouter-mode               | Accept Router Advertisements if forwarding is disabled.     |
| always                      | Accept Router Advertisements even if forwarding is enabled. |

### Default value

never

## accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router.

```
vrouter running config# system network-stack ipv6
vrouter running ipv6# accept-redirects true|false
```

### Default value

false

## accept-source-route

Accept packets with source route option.

```
vrouter running config# system network-stack ipv6
vrouter running ipv6# accept-source-route true|false
```

### Default value

false

## router-solicitations

Number of Router Solicitations to send until assuming no routers are present.

```
vrouter running config# system network-stack ipv6
vrouter running ipv6# router-solicitations <int16>
```

### Default value

-1

## use-temporary-addresses

Preference for Privacy Extensions (RFC4941). Not applied to point-to-point and loopback devices (always 0).

```
vrouter running config# system network-stack ipv6
vrouter running ipv6# use-temporary-addresses USE-TEMPORARY-ADDRESSES
```

| USE-TEMPORARY-ADDRESSES values | Description   |
|--------------------------------|---|
| never                          | Disable Privacy Extensions, i.e. use the public address, subnet prefix/interface id, where interface id is always the same. |
| prefer-public-addresses        | Enable Privacy Extensions, but prefer public addresses over temporary addresses.  |
| always                         | Enable Privacy Extensions and prefer temporary addresses over public addresses.   |

### Default value

never

## neighbor

Neighbor advanced configuration.

```
vrouter running config# system network-stack neighbor
```

## ipv4-max-entries

Maximum number of IPv4 neighbors.

```
vrouter running config# system network-stack neighbor
vrouter running neighbor# ipv4-max-entries <uint32>
```

### ipv6-max-entries

Maximum number of IPv6 neighbors.

```
vrouter running config# system network-stack neighbor  
vrouter running neighbor# ipv6-max-entries <uint32>
```

### ipv4-base-reachable-time

Time during which an IPv4 neighbor entry stays reachable.

```
vrouter running config# system network-stack neighbor  
vrouter running neighbor# ipv4-base-reachable-time <uint32>
```

### ipv6-base-reachable-time

Time during which an IPv6 neighbor entry stays reachable.

```
vrouter running config# system network-stack neighbor  
vrouter running neighbor# ipv6-base-reachable-time <uint32>
```

### contrack

Contrack advanced configuration.

```
vrouter running config# system network-stack contrack
```

### max-entries

Maximum number of Netfilter contracks.

```
vrouter running config# system network-stack contrack  
vrouter running contrack# max-entries <uint32>
```

### tcp-timeout-close

Conntrack TCP timeout close.

```
vrouter running config# system network-stack conntrack
vrouter running conntrack# tcp-timeout-close <uint32>
```

### tcp-timeout-close-wait

Conntrack TCP timeout close wait.

```
vrouter running config# system network-stack conntrack
vrouter running conntrack# tcp-timeout-close-wait <uint32>
```

### tcp-timeout-established

Conntrack TCP timeout established.

```
vrouter running config# system network-stack conntrack
vrouter running conntrack# tcp-timeout-established <uint32>
```

### tcp-timeout-fin-wait

Conntrack TCP timeout fin wait.

```
vrouter running config# system network-stack conntrack
vrouter running conntrack# tcp-timeout-fin-wait <uint32>
```

### tcp-timeout-last-ack

Conntrack TCP timeout last ack.

```
vrouter running config# system network-stack conntrack
vrouter running conntrack# tcp-timeout-last-ack <uint32>
```

### tcp-timeout-max-retrans

Conntrack TCP timeout max retrans.

```
vrouter running config# system network-stack conntrack  
vrouter running conntrack# tcp-timeout-max-retrans <uint32>
```

### tcp-timeout-syn-recv

Conntrack TCP timeout syn recv.

```
vrouter running config# system network-stack conntrack  
vrouter running conntrack# tcp-timeout-syn-recv <uint32>
```

### tcp-timeout-syn-sent

Conntrack TCP timeout syn sent.

```
vrouter running config# system network-stack conntrack  
vrouter running conntrack# tcp-timeout-syn-sent <uint32>
```

### tcp-timeout-time-wait

Conntrack TCP timeout time wait.

```
vrouter running config# system network-stack conntrack  
vrouter running conntrack# tcp-timeout-time-wait <uint32>
```

### tcp-timeout-unacknowledged

Conntrack TCP timeout unacknowledged.

```
vrouter running config# system network-stack conntrack  
vrouter running conntrack# tcp-timeout-unacknowledged <uint32>
```

### udp-timeout

Conntrack UDP timeout.

```
vrouter running config# system network-stack conntrack
vrouter running conntrack# udp-timeout <uint32>
```

### udp-timeout-stream

Conntrack UDP timeout stream.

```
vrouter running config# system network-stack conntrack
vrouter running conntrack# udp-timeout-stream <uint32>
```

### installed-image (state only)

The list of installed images.

### version (state only)

The version of the image.

```
vrouter> show state system installed-image <string> version
```

### current (state only)

The image is currently booted.

```
vrouter> show state system installed-image <string> current
```

### default (state only)

The image is booted by default.

```
vrouter> show state system installed-image <string> default
```

### next (state only)

The next reboot will use this image.

```
vrouter> show state system installed-image <string> next
```

## 3.2.5 cloud-init

Cloud-init configuration.

```
vrouter running config# system cloud-init
```

### enabled

Enable or disable cloud-init.

```
vrouter running config# system cloud-init  
vrouter running cloud-init# enabled true|false
```

### datasource (state only)

The selected datasource, if any.

```
vrouter> show state system cloud-init datasource
```

## 3.2.6 license

License configuration.

```
vrouter running config# system license
```

### enabled

Enable or disable the license.

```
vrouter running config# system license  
vrouter running license# enabled true|false
```

### Default value

true

**valid (state only)**

True if the license is valid.

```
vrouter> show state system license valid
```

**state (state only)**

The license state.

```
vrouter> show state system license state
```

**activation-type (state only)**

How the license was activated.

```
vrouter> show state system license activation-type
```

**license-type (state only)**

The license type.

```
vrouter> show state system license license-type
```

**short-license-type (state only)**

A shorter version of the license type.

```
vrouter> show state system license short-license-type
```

**subscription-end-date (state only)**

The subscription or evaluation end date.

```
vrouter> show state system license subscription-end-date
```

**support-type (state only)**

The type of support (none, standard or extended).

```
vrouter> show state system license support-type
```

**support-end-date (state only)**

The support end date.

```
vrouter> show state system license support-end-date
```

**online**

Configure an online license.

```
vrouter running config# system license online
```

**serial (mandatory)**

The license serial number used for this machine.

```
vrouter running config# system license online
vrouter running online# serial <string>
```

**vrf**

The VRF to use for connecting to the license server.

```
vrouter running config# system license online
vrouter running online# vrf VRF
```

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

**Default value**

main

### proxy-host

The hostname or address of an HTTP proxy to use for connecting to the online license server.

```
vrouter running config# system license online
vrouter running online# proxy-host PROXY-HOST
```

| PROXY-Host values | Description  |
|-------------------|--|
| <A.B.C.D>         | IPv4 address.  |
| <X:X::X:X>        | IPv6 address.  |
| <host-name>       | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |

### proxy-port

The TCP port of the HTTP proxy server.

```
vrouter running config# system license online
vrouter running online# proxy-port PROXY-PORT
```

|            |   |
|------------|---|
| PROXY-PORT | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|------------|---|

### proxy-user

The username for connecting to the HTTP proxy server.

```
vrouter running config# system license online  
vrouter running online# proxy-user <string>
```

### proxy-password

The password for connecting to the HTTP proxy server.

```
vrouter running config# system license online  
vrouter running online# proxy-password <string>
```

### export-analytics

If activated, exports information necessary for finding the instance on the licensing portal. It is recommended to leave this enabled.

```
vrouter running config# system license online  
vrouter running online# export-analytics true|false
```

#### Default value

true

### use-ipv6-dns

If activated, the licensing server will be accessed using IPv6 instead of IPv4.

```
vrouter running config# system license online  
vrouter running online# use-ipv6-dns true|false
```

#### Default value

false

**connected (state only)**

True if the cloud license server is reachable.

```
vrouter> show state system license online connected
```

**last-healthcheck-date (state only)**

The last contact with license server date.

```
vrouter> show state system license online last-healthcheck-date
```

**lease-end-date (state only)**

The end of the license lease date.

```
vrouter> show state system license online lease-end-date
```

**current-activations (state only)**

How many devices already hold this license.

```
vrouter> show state system license online current-activations
```

**allowed-activations (state only)**

How many devices can hold this license at a time.

```
vrouter> show state system license online allowed-activations
```

**computer-id (state only)**

This computer ID.

```
vrouter> show state system license online computer-id
```

## offline-certificate

Configure an offline license using a certificate.

```
vrouter running config# system license offline-certificate
```

### serial (mandatory)

The license serial number of a license file imported with the `license certificate import` command. After the first successful commit, the certificate is deleted.

```
vrouter running config# system license offline-certificate  
vrouter running offline-certificate# serial <string>
```

### computer-id (state only)

This computer ID.

```
vrouter> show state system license offline-certificate computer-id
```

### license-certificate (state only)

License certificates imported on the system.

```
vrouter> show state system license offline-certificate license-certificate
```

## offline

Configure an offline license using a file.

```
vrouter running config# system license offline
```

### serial (mandatory)

The license serial number of a license file imported with the `license-file import` command.

```
vrouter running config# system license offline  
vrouter running offline# serial <string>
```

**license-file (state only)**

License files imported on the system.

```
vrouter> show state system license offline license-file
```

**throughput (state only)**

Network throughput.

**allowed (state only)**

The allowed throughput.

```
vrouter> show state system license throughput allowed
```

**used (state only)**

The throughput currently in use.

```
vrouter> show state system license throughput used
```

**cgnat-contracks (state only)**

CG-NAT contracks.

**allowed (state only)**

The number of CG-NAT contracks allowed.

```
vrouter> show state system license cgnat-contracks allowed
```

**used (state only)**

The number of CG-NAT contracks currently in use.

```
vrouter> show state system license cgnat-contracks used
```

**ipsec-tunnels (state only)**

IPsec tunnels.

**allowed (state only)**

The number of IPsec tunnels allowed.

```
vrouter> show state system license ipsec-tunnels allowed
```

**used (state only)**

The number of IPsec tunnels currently in use.

```
vrouter> show state system license ipsec-tunnels used
```

**3.2.7 auth**

Configuration data for local users.

```
vrouter running config# system auth
```

**user**

List of local users on the system.

```
vrouter running config# system auth user <string>
```

|                             |  |
|-----------------------------|--|
| <code>&lt;string&gt;</code> | The user name string identifying this entry. |
|-----------------------------|--|

**role (mandatory)**

The role of the user.

```
vrouter running config# system auth user <string>
vrouter running user <string># role ROLE
```

| ROLE values | Description   |
|-------------|---|
| viewer      | The user can view configuration and state and run standard commands. However, he/she cannot edit the configuration, read protected config/state nodes (such as passwords) nor run privileged commands (such as reboot, poweroff, etc.). |
| admin       | The user can view all configuration and state, including protected nodes (such as password). He/she may edit the configuration and run any command including privileged ones (such as reboot, poweroff, etc.).                          |

## password

The user password, supplied as a hashed value using the notation described in the definition of the crypt-hash type.

```
vrouter running config# system auth user <string>
vrouter running user <string># password PASSWORD
```

**PASS**-The crypt-hash type is used to store passwords using a hash function. The algorithms for applying the **WORD** hash function and encoding the result are implemented in various UNIX systems as the function crypt(3). A value of this type matches one of the forms: `$0$<clear text password> $<id>$<salt>$<password hash> $<id>$<parameter>$<salt>$<password hash>` The ‘\$0\$’ prefix signals that the value is clear text. When such a value is received by the server, a hash value is calculated, and the string ‘\$<id>\$<salt>’ or ‘\$<id>\$<parameter>\$<salt>\$’ is prepended to the result. This value is stored in the configuration data store. If a value starting with ‘\$<id>\$’, where <id> is not ‘0’, is received, the server knows that the value already represents a hashed value and stores it ‘as is’ in the data store. When a server needs to verify a password given by a user, it finds the stored password hash string for that user, extracts the salt, and calculates the hash with the salt and given password as input. If the calculated hash value is the same as the stored value, the password given by the client is accepted. This type defines the following hash functions: id | hash function | feature —+—————+————— 1 | MD5 | crypt-hash-md5 5 | SHA-256 | crypt-hash-sha-256 6 | SHA-512 | crypt-hash-sha-512 The server indicates support for the different hash functions by advertising the corresponding feature.

## authorized-key

A public SSH key for this user in the OpenSSH format. This key is allowed for SSH authentication without a password to both the NETCONF and SSH servers. You may use the ssh-keygen utility to generate a new key-pair and paste the contents of the \*.pub file (the public key) here.

```
vrouter running config# system auth user <string>
vrouter running user <string># authorized-key <string>
```

### 3.2.8 aaa

---

**Note:** requires a Turbo Router Network License.

---

Configuration data for aaa servers.

```
vrouter running config# system aaa
```

#### tacacs

List of tacacs servers on the system.

```
vrouter running config# system aaa tacacs <uint32>
```

|          |  |
|----------|--|
| <uint32> | Order for TACACS+ servers. They will be reached by increasing order value. |
|----------|--|

#### address (mandatory)

TACACS+ server IPv4 or IPv6 address. It has to be accessible from vrf 'main'.

```
vrouter running config# system aaa tacacs <uint32>
vrouter running tacacs <uint32># address ADDRESS
```

| ADDRESS values | Description      |
|----------------|------------------|
| <A.B.C.D>      | An IPv4 address. |
| <X:X::X:X>     | An IPv6 address. |

#### port

Port number to reach the TACACS server.

```
vrouter running config# system aaa tacacs <uint32>
vrouter running tacacs <uint32># port <uint16>
```

#### Default value

49

**secret (mandatory)**

TACACS+ client/server shared secret.

```
vrouter running config# system aaa tacacs <uint32>
vrouter running tacacs <uint32># secret <string>
```

**timeout**

Timeout before trying to reach another TACACS+ server.

```
vrouter running config# system aaa tacacs <uint32>
vrouter running tacacs <uint32># timeout <uint8>
```

**Default value**

3

**3.2.9 vrf**

Vrf list.

```
vrouter running config# vrf <vrf>
```

| <vrf> values | Description   |
|--------------|---------------|
| main         | The main vrf. |
| <string>     | The vrf name. |

**3.2.10 ssh-server**

Top-level container for ssh server.

```
vrouter running config# vrf <vrf> ssh-server
```

**enabled**

Enable or disable the ssh server.

```
vrouter running config# vrf <vrf> ssh-server
vrouter running ssh-server# enabled true|false
```

**Default value**

true

## address

The IP address of the interface to listen on. The SSH server will listen on all interfaces if no value is specified.

```
vrouter running config# vrf <vrf> ssh-server
vrouter running ssh-server# address ADDRESS
```

| ADDRESS values | Description      |
|----------------|------------------|
| <A.B.C.D>      | An IPv4 address. |
| <X:X::X:X>     | An IPv6 address. |

## port

The local port number on this interface the SSH server listens on.

```
vrouter running config# vrf <vrf> ssh-server
vrouter running ssh-server# port PORT
```

|      |   |
|------|---|
| PORT | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|------|---|

## Default value

22

## permit-root-login

Allow to login as root.

```
vrouter running config# vrf <vrf> ssh-server
vrouter running ssh-server# permit-root-login PERMIT-ROOT-LOGIN
```

| PERMIT-ROOT-LOGIN values | Description                 |
|--------------------------|-----------------------------|
| yes                      | Permit root login.          |
| no                       | Doesn't permit root login.  |
| prohibit-password        | Prohibit login by password. |
| <string>                 | No description.             |

## Default value

yes

### 3.2.11 netconf-server

Configuration data for NETCONF server.

```
vrouter running config# vrf <vrf> netconf-server
```

#### enabled

Enable or disable NETCONF server. If no addresses are specified, NETCONF will listen on all IPv4 and IPv6 addresses on port 830.

```
vrouter running config# vrf <vrf> netconf-server
vrouter running netconf-server# enabled true|false
```

#### Default value

true

#### idle-timeout

Specifies the maximum number of seconds that a NETCONF session may remain idle. A NETCONF session will be dropped if it is idle for an interval longer than this number of seconds. If set to zero, then the server will never drop a session because it is idle. Sessions that have a notification subscription active are never dropped.

```
vrouter running config# vrf <vrf> netconf-server
vrouter running netconf-server# idle-timeout <uint16>
```

#### Default value

3600

#### address

List of addresses on which to listen to NETCONF clients.

```
vrouter running config# vrf <vrf> netconf-server
vrouter running netconf-server# address <address> port PORT description <string>
```

| <address> values | Description      |
|------------------|------------------|
| <A.B.C.D>        | An IPv4 address. |
| <X:X::X:X>       | An IPv6 address. |

## port

The port number to listen on (default: 830).

```
port PORT
```

|      |   |
|------|---|
| PORT | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|------|---|

### Default value

830

## description

NETCONF listen endpoint description.

```
description <string>
```

## 3.2.12 dns

Enclosing container for DNS resolver data.

```
vrouter running config# vrf <vrf> dns
```

## search

An ordered list of domains to search when resolving a host name.

```
vrouter running config# vrf <vrf> dns
vrouter running dns# search SEARCH
```

**SEARCH** The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492.

## server

List of the DNS servers that the resolver should query. When the resolver is invoked by a calling application, it sends the query to the first name server in this list. If no response has been received within ‘timeout’ seconds, the resolver continues with the next server in the list. If no response is received from any server, the resolver continues with the first server again. When the resolver has traversed the list ‘attempts’ times without receiving any response, it gives up and returns an error to the calling application. Implementations MAY limit the number of entries in this list.

```
vrouter running config# vrf <vrf> dns
vrouter running dns# server <server>
```

| <server> values | Description      |
|-----------------|------------------|
| <A.B.C.D>       | An IPv4 address. |
| <X:X::X:X>      | An IPv6 address. |

## proxy

DNS proxy configuration.

```
vrouter running config# vrf <vrf> dns proxy
```

## enabled

Enable or disable DNS proxy. By default, DNS proxy listens to requests on all networks and forwards them to local DNS servers (configured statically or obtained through DHCP).

```
vrouter running config# vrf <vrf> dns proxy
vrouter running proxy# enabled true|false
```

## Default value

true

## listen-to

Configure networks on which to listen to DNS requests. If not specified, DNS proxy listens to all networks.

```
vrouter running config# vrf <vrf> dns proxy
vrouter running proxy# listen-to LISTEN-T0
```

| LISTEN-T0 values | Description                            |
|------------------|--|
| <A.B.C.D/M>      | An IPv4 prefix: address and CIDR mask. |
| <X:X::X:X/M>     | An IPv6 prefix: address and CIDR mask. |

## forward

Configure name servers to forward the DNS requests to. If not specified, requests are forwarded to local DNS servers (configured statically or obtained through DHCP).

```
vrouter running config# vrf <vrf> dns proxy forward
```

## server

The address of the DNS servers, can be either IPv4 or IPv6.

```
vrouter running config# vrf <vrf> dns proxy forward
vrouter running forward# server SERVER
```

| SERVER values | Description      |
|---------------|------------------|
| <A.B.C.D>     | An IPv4 address. |
| <X:X::X:X>    | An IPv6 address. |

## local

Forward DNS requests to local DNS servers (configured statically or obtained through DHCP).

```
vrouter running config# vrf <vrf> dns proxy forward
vrouter running forward# local
```

## dns64

DNS64 configuration.

```
vrouter running config# vrf <vrf> dns proxy dns64 <dns64>
```

|         |  |
|---------|--|
| <dns64> | An IPv6 prefix: address and CIDR mask. |
|---------|--|

## client

Clients IPv6 addresses.

```
vrouter running config# vrf <vrf> dns proxy dns64 <dns64>
vrouter running dns64 <dns64># client CLIENT
```

|        |  |
|--------|--|
| CLIENT | An IPv6 prefix: address and CIDR mask. |
|--------|--|

## exclude

IPv6 addresses to exclude.

```
vrouter running config# vrf <vrf> dns proxy dns64 <dns64>  
vrouter running dns64 <dns64># exclude EXCLUDE
```

|         |  |
|---------|--|
| EXCLUDE | An IPv6 prefix: address and CIDR mask. |
|---------|--|

## mapped

IPv4 prefixes to be map in the corresponding A resource record set. If not defined it defaults to any.

```
vrouter running config# vrf <vrf> dns proxy dns64 <dns64>  
vrouter running dns64 <dns64># mapped [not] PREFIX
```

## not

Do not map the following set of IPv4 prefixes.

```
not
```

## PREFIX

IPv4 prefixes to be mapped or not.

```
PREFIX
```

|        |  |
|--------|--|
| PREFIX | An IPv4 prefix: address and CIDR mask. |
|--------|--|

### 3.2.13 lldp

---

**Note:** requires a Turbo Router Network License.

---

Top-level container for LLDP configuration and state data.

```
vrouter running config# vrf <vrf> lldp
```

## enabled

System level state of the LLDP protocol.

```
vrouter running config# vrf <vrf> lldp
vrouter running lldp# enabled true|false
```

## Default value

true

## hello-timer

System level hello timer for the LLDP protocol.

```
vrouter running config# vrf <vrf> lldp
vrouter running lldp# hello-timer <uint64>
```

## system-name

The system name field shall contain an alpha-numeric string that indicates the system's administratively assigned name. The system name should be the system's fully qualified domain name. If implementations support IETF RFC 3418, the sysName object should be used for this field.

```
vrouter running config# vrf <vrf> lldp
vrouter running lldp# system-name <string>
```

## system-description

The system description field shall contain an alpha-numeric string that is the textual description of the network entity. The system description should include the full name and version identification of the system's hardware type, software operating system, and networking software. If implementations support IETF RFC 3418, the sysDescr object should be used for this field.

```
vrouter running config# vrf <vrf> lldp
vrouter running lldp# system-description <string>
```

## management-address

The Management Address is a mandatory TLV which identifies a network address associated with the local LLDP agent, which can be used to reach the agent on the port identified in the Port ID TLV.

```
vrouter running config# vrf <vrf> lldp
vrouter running lldp# management-address <string>
```

## chassis-id (state only)

The Chassis ID is a mandatory TLV which identifies the chassis component of the endpoint identifier associated with the transmitting LLDP agent.

```
vrouter> show state vrf <vrf> lldp chassis-id
```

## chassis-id-type (state only)

This field identifies the format and source of the chassis identifier string. It is an enumerator defined by the LldpChassisIdSubtype object from IEEE 802.1AB MIB.

```
vrouter> show state vrf <vrf> lldp chassis-id-type
```

## interface

List of interfaces on which LLDP is enabled / available.

```
vrouter running config# vrf <vrf> lldp interface <interface>
```

|             |                    |
|-------------|--------------------|
| <interface> | An interface name. |
|-------------|--------------------|

## enabled

Enable or disable the LLDP protocol on the interface.

```
vrouter running config# vrf <vrf> lldp interface <interface>
vrouter running interface <interface># enabled true|false
```

## Default value

true

**counters (state only)**

LLDP counters on each interface.

**frame-in (state only)**

The number of lldp frames received.

```
vrouter> show state vrf <vrf> lldp interface <interface> counters frame-in
```

**frame-out (state only)**

The number of frames transmitted out.

```
vrouter> show state vrf <vrf> lldp interface <interface> counters frame-out
```

**frame-discard (state only)**

The number of LLDP frames received and discarded.

```
vrouter> show state vrf <vrf> lldp interface <interface> counters frame-discard
```

**tlv-discard (state only)**

The number of TLV frames received and discarded.

```
vrouter> show state vrf <vrf> lldp interface <interface> counters tlv-discard
```

**neighbor (state only)**

List of LLDP neighbors.

### port-id (state only)

The Port ID is a mandatory TLV which identifies the port component of the endpoint identifier associated with the transmitting LLDP agent. If the specified port is an IEEE 802.3 Repeater port, then this TLV is optional.

```
vrouter> show state vrf <vrf> lldp interface <interface> neighbor id <string> port-id
```

### port-id-type (state only)

This field identifies the format and source of the port identifier string. It is an enumerator defined by the PtopoPortIdType object from RFC2922.

```
vrouter> show state vrf <vrf> lldp interface <interface> neighbor id <string> port-id-  
↪type
```

### port-description (state only)

The binary string containing the actual port identifier for the port which this LLDP PDU was transmitted. The source and format of this field is defined by PtopoPortId from RFC2922.

```
vrouter> show state vrf <vrf> lldp interface <interface> neighbor id <string> port-  
↪description
```

### management-address (state only)

The Management Address is a mandatory TLV which identifies a network address associated with the local LLDP agent, which can be used to reach the agent on the port identified in the Port ID TLV.

```
vrouter> show state vrf <vrf> lldp interface <interface> neighbor id <string> ↪  
↪management-address
```

### system-name (state only)

The system name field shall contain an alpha-numeric string that indicates the system's administratively assigned name. The system name should be the system's fully qualified domain name. If implementations support IETF RFC 3418, the sysName object should be used for this field.

```
vrouter> show state vrf <vrf> lldp interface <interface> neighbor id <string> system-  
↪name
```

### system-description (state only)

The system description field shall contain an alpha-numeric string that is the textual description of the network entity. The system description should include the full name and version identification of the system's hardware type, software operating system, and networking software. If implementations support IETF RFC 3418, the sysDescr object should be used for this field.

```
vrouter> show state vrf <vrf> lldp interface <interface> neighbor id <string> system-  
↪description
```

### chassis-id (state only)

The Chassis ID is a mandatory TLV which identifies the chassis component of the endpoint identifier associated with the transmitting LLDP agent.

```
vrouter> show state vrf <vrf> lldp interface <interface> neighbor id <string> chassis-  
↪id
```

### chassis-id-type (state only)

This field identifies the format and source of the chassis identifier string. It is an enumerator defined by the LldpChassisIdSubtype object from IEEE 802.1AB MIB.

```
vrouter> show state vrf <vrf> lldp interface <interface> neighbor id <string> chassis-  
↪id-type
```

### capability (state only)

List of LLDP system capabilities advertised by the neighbor.

### enabled (state only)

Indicates whether the corresponding system capability is enabled on the neighbor.

```
vrouter> show state vrf <vrf> lldp interface <interface> neighbor id <string> ↪  
↪capability <capability> enabled
```

**counters (state only)**

Global LLDP counters.

**frame-in (state only)**

The number of lldp frames received.

```
vrouter> show state vrf <vrf> lldp counters frame-in
```

**frame-out (state only)**

The number of frames transmitted out.

```
vrouter> show state vrf <vrf> lldp counters frame-out
```

**frame-discard (state only)**

The number of LLDP frames received and discarded.

```
vrouter> show state vrf <vrf> lldp counters frame-discard
```

**tlv-discard (state only)**

The number of TLV frames received and discarded.

```
vrouter> show state vrf <vrf> lldp counters tlv-discard
```

**tlv-accepted (state only)**

The number of valid TLVs received.

```
vrouter> show state vrf <vrf> lldp counters tlv-accepted
```

### entries-aged-out (state only)

The number of entries aged out due to timeout.

```
vrouter> show state vrf <vrf> lldp counters entries-aged-out
```

### 3.2.14 kpi

---

**Note:** requires a Turbo Router Network License.

---

KPI configuration for interface and telegraf agent.

```
vrouter running config# vrf <vrf> kpi
```

#### interface

Tell which interfaces should be polled by network-nic-\* services in this vrf. Default is to take the ones polled by the fast path.

```
vrouter running config# vrf <vrf> kpi  
vrouter running kpi# interface INTERFACE
```

|           |                    |
|-----------|--------------------|
| INTERFACE | An interface name. |
|-----------|--------------------|

### 3.2.15 telegraf

---

**Note:** requires a Turbo Router Network License.

---

Telegraf configuration.

```
vrouter running config# vrf <vrf> kpi telegraf
```

**enabled**

Enable or disable telegraf.

```
vrouter running config# vrf <vrf> kpi telegraf
vrouter running telegraf# enabled true|false
```

**Default value**

true

**interval**

Default data collection interval in seconds.

```
vrouter running config# vrf <vrf> kpi telegraf
vrouter running telegraf# interval <uint16>
```

**Default value**

10

**influxdb-output**

Configure an InfluxDB server.

```
vrouter running config# vrf <vrf> kpi telegraf
vrouter running telegraf# influxdb-output url <influxdb-output> database <string> \
... username <string> password <string> insecure-skip-verify
```

| <influxdb-output> values   | Description          |
|----------------------------|----------------------|
| <udp://host[:port]>        | An UDP URL.          |
| <udp://[host6][:port]>     | An IPv6 UDP URL.     |
| <http[s]://host[:port]>    | An HTTP(S) URL.      |
| <http[s]://[host6][:port]> | An IPv6 HTTP(S) URL. |

**database (mandatory)**

The target database for metrics (telegraf will create it if not exists).

```
database <string>
```

### username

The username to connect to InfluxDB.

```
username <string>
```

### password

The password to connect to InfluxDB.

```
password <string>
```

### insecure-skip-verify

Use SSL but skip chain and host verification.

```
insecure-skip-verify
```

## 3.2.16 tracker

---

**Note:** requires a Turbo Router Network License.

---

Track IP addresses.

```
vrouter running config# tracker
```

### bfd

Configure a BFD tracker session.

```
vrouter running config# tracker bfd <bfd>
```

|       |                  |
|-------|------------------|
| <bfd> | An tracker name. |
|-------|------------------|

**type**

Session type.

```
vrouter running config# tracker bfd <bfd>
vrouter running bfd <bfd># type TYPE
```

| TYPE values | Description         |
|-------------|---------------------|
| single-hop  | Single-hop session. |
| multi-hop   | Multi-hop session.  |

**Default value**

single-hop

**source**

Local IP address.

```
vrouter running config# tracker bfd <bfd>
vrouter running bfd <bfd># source SOURCE
```

| SOURCE values | Description      |
|---------------|------------------|
| <A.B.C.D>     | An IPv4 address. |
| <X:X::X:X>    | An IPv6 address. |

**address (mandatory)**

IP address of the peer.

```
vrouter running config# tracker bfd <bfd>
vrouter running bfd <bfd># address ADDRESS
```

| ADDRESS values | Description      |
|----------------|------------------|
| <A.B.C.D>      | An IPv4 address. |
| <X:X::X:X>     | An IPv6 address. |

## interface

Interface to use to contact peer.

```
vrouter running config# tracker bfd <bfd>
vrouter running bfd <bfd># interface INTERFACE
```

|           |                    |
|-----------|--------------------|
| INTERFACE | An interface name. |
|-----------|--------------------|

## vrf (mandatory)

VRF name.

```
vrouter running config# tracker bfd <bfd>
vrouter running bfd <bfd># vrf VRF
```

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

## echo-mode

Use echo packets to detect failures.

```
vrouter running config# tracker bfd <bfd>
vrouter running bfd <bfd># echo-mode true|false
```

## detection-multiplier

Local session detection multiplier.

```
vrouter running config# tracker bfd <bfd>
vrouter running bfd <bfd># detection-multiplier <uint8>
```

### Default value

3

**desired-transmission-interval**

Minimum desired control packet transmission interval.

```
vrouter running config# tracker bfd <bfd>
vrouter running bfd <bfd># desired-transmission-interval <uint32>
```

**Default value**

300000

**required-receive-interval**

Minimum required control packet receive interval (use disable to not receive any control packet).

```
vrouter running config# tracker bfd <bfd>
vrouter running bfd <bfd># required-receive-interval REQUIRED-RECEIVE-INTERVAL
```

| REQUIRED-RECEIVE-INTERVAL values | Description  |
|----------------------------------|--|
| <uint32>                         | No description.  |
| disable                          | This system will not receive any periodic BFD control packets. |

**Default value**

300000

**desired-echo-transmission-interval**

Minimum desired control packet transmission interval.

```
vrouter running config# tracker bfd <bfd>
vrouter running bfd <bfd># desired-echo-transmission-interval <uint32>
```

**discriminator (state only)**

Local session identifier.

```
vrouter> show state tracker bfd <bfd> discriminator
```

**state (state only)**

Local session state.

```
vrouter> show state tracker bfd <bfd> state
```

**diagnostic (state only)**

Local session diagnostic.

```
vrouter> show state tracker bfd <bfd> diagnostic
```

**last-down-time (state only)**

Time and date of the last time session was down (in seconds).

```
vrouter> show state tracker bfd <bfd> last-down-time
```

**last-up-time (state only)**

Time and date of the last time session was up (in seconds).

```
vrouter> show state tracker bfd <bfd> last-up-time
```

**session-down-count (state only)**

Amount of time the session went down.

```
vrouter> show state tracker bfd <bfd> session-down-count
```

**session-up-count (state only)**

Amount of time the session went up.

```
vrouter> show state tracker bfd <bfd> session-up-count
```

**control-packet-input-count (state only)**

Amount of control packets received.

```
vrouter> show state tracker bfd <bfd> control-packet-input-count
```

**control-packet-output-count (state only)**

Amount of control packets sent.

```
vrouter> show state tracker bfd <bfd> control-packet-output-count
```

**echo-packet-input-count (state only)**

Amount of echo packets received.

```
vrouter> show state tracker bfd <bfd> echo-packet-input-count
```

**echo-packet-output-count (state only)**

Amount of echo packets sent.

```
vrouter> show state tracker bfd <bfd> echo-packet-output-count
```

**zebra-notification-count (state only)**

Amount of zebra notifications.

```
vrouter> show state tracker bfd <bfd> zebra-notification-count
```

**remote (state only)**

BFD remote operational state data.

**discriminator (state only)**

Remote session identifier.

```
vrouter> show state tracker bfd <bfd> remote discriminator
```

**diagnostic (state only)**

Local session diagnostic.

```
vrouter> show state tracker bfd <bfd> remote diagnostic
```

**multiplier (state only)**

Remote session detection multiplier.

```
vrouter> show state tracker bfd <bfd> remote multiplier
```

**negotiated (state only)**

BFD negotiated operational state data.

**transmission-interval (state only)**

Negotiated transmit interval.

```
vrouter> show state tracker bfd <bfd> negotiated transmission-interval
```

**receive-interval (state only)**

Negotiated receive interval.

```
vrouter> show state tracker bfd <bfd> negotiated receive-interval
```

**echo-transmission-interval (state only)**

Negotiated echo transmit interval.

```
vrouter> show state tracker bfd <bfd> negotiated echo-transmission-interval
```

**icmp**


---

**Note:** requires a Turbo Router Network License.

---

List of tracked addresses using ICMP echo requests.

```
vrouter running config# tracker
vrouter running tracker# icmp <icmp> address ADDRESS vrf VRF source SOURCE \
... interface INTERFACE dhcp-interface DHCP-INTERFACE gateway GATEWAY period <uint16> \
... threshold <uint8> total <uint8> packet-size <uint16> packet-tos <uint8> timeout
↵<uint16>
```

|        |                  |
|--------|------------------|
| <icmp> | An tracker name. |
|--------|------------------|

**address**

The host to track.

```
address ADDRESS
```

| ADDRESS values | Description  |
|----------------|--|
| <A.B.C.D>      | An IPv4 address.   |
| <X:X::X:X>     | An IPv6 address.   |
| <host-name>    | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |

### vrf (mandatory)

The vrf in which the ping must be sent. Default is the current netns.

```
vrf VRF
```

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

### source

Source address in the ping packet.

```
source SOURCE
```

| SOURCE values | Description      |
|---------------|------------------|
| <A.B.C.D>     | An IPv4 address. |
| <X:X::X:X>    | An IPv6 address. |

### interface

The interface to bind the tracker to.

```
interface INTERFACE
```

|           |                    |
|-----------|--------------------|
| INTERFACE | An interface name. |
|-----------|--------------------|

### dhcp-interface

The address, gateway and source will be taken from DHCP on this interface unless explicitly specified in the tracker.

```
dhcp-interface DHCP-INTERFACE
```

|                |                    |
|----------------|--------------------|
| DHCP-INTERFACE | An interface name. |
|----------------|--------------------|

### gateway

The gateway to use to send the packet.

```
gateway GATEWAY
```

| GATEWAY values | Description  |
|----------------|--|
| <A.B.C.D>      | IPv4 address.  |
| <X:X::X:X>     | IPv6 address.  |
| <host-name>    | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |

**period**

Time between each ping.

```
period <uint16>
```

**Default value**

500

**threshold**

Number of successful pings among <total> to consider peer as reachable.

```
threshold <uint8>
```

**Default value**

1

**total**

Check the threshold among this number of last pings to consider peer as reachable.

```
total <uint8>
```

**Default value**

1

**packet-size**

Packet size.

```
packet-size <uint16>
```

**Default value**

100

### packet-tos

ToS to apply to the packet.

```
packet-tos <uint8>
```

#### Default value

192

### timeout

Time during which a ping reply is considered as valid. If unset, it timeouts after a ping period.

```
timeout <uint16>
```

### state (state only)

Status of the last ping.

```
vrouter> show state tracker icmp <icmp> state
```

### diagnostic (state only)

Local session diagnostic.

```
vrouter> show state tracker icmp <icmp> diagnostic
```

## 3.2.17 nat

---

**Note:** requires a Turbo Router Network License.

---

NAT configuration.

```
vrouter running config# vrf <vrf> nat
```

## source-rule

A rule to change the source address/port of outgoing packets.

```

vrouter running config# vrf <vrf> nat
vrouter running nat# source-rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... source \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... outbound-interface [not] <string> \
... translate-to map MAP output-address \
... address VALUE port PORT \
... port-range START END \
... address-range START END port PORT \
... port-range START END

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

## description

A comment to describe the rule.

```
description <string>
```

## protocol

Match a protocol.

```
protocol [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description    |
|--------------|----------------|
| tcp          | TCP protocol.  |
| udp          | UDP protocol.  |
| icmp         | ICMP protocol. |
| all          | All protocols. |

**destination**

Match a destination attribute.

```
destination \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

**address**

Match this destination address or prefix.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | prefix: address and CIDR mask.   |

**port**

Match this destination port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**source**

Match a source attribute.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

**address**

Match this source address or prefix.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE                            | Description  |
|----------------------------------|--|
| val-<br>ues<br><br><domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>                        | IPv4 address.  |
| <A.B.C/DIM>                      | prefix: address and CIDR mask.   |

**port**

Match this source port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**outbound-interface**

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

**not**

Invert the match.

```
not
```

**<string> (mandatory)**

The interface to match.

```
<string>
```

**translate-to**

Translate to.

```
translate-to map MAP output-address \  
  address VALUE port PORT \  
  port-range START END \  
  address-range START END port PORT \  
  port-range START END
```

**map**

Translate a whole network of addresses onto another network of addresses. All ‘one’ bits in the mask are filled in from the new address. All bits that are zero in the mask are filled in from the original address.

```
map MAP
```

|     |  |
|-----|--|
| MAP | An IPv4 prefix: address and CIDR mask. |
|-----|--|

**output-address**

Translate to the address found on the outgoing interface.

```
output-address
```

**address**

Translate to an address and port/port range.

```
address VALUE port PORT \  
    port-range START END
```

**VALUE (mandatory)**

Translate to an address.

```
VALUE
```

|       |                  |
|-------|------------------|
| VALUE | An IPv4 address. |
|-------|------------------|

**port**

Translate to a port.

```
port PORT
```

|      |   |
|------|---|
| PORT | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|------|---|

## port-range

Translate to a port range.

```
port-range START END
```

### START (mandatory)

Port range start.

```
START
```

|       |   |
|-------|---|
| START | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

### END (mandatory)

Port range end.

```
END
```

|     |   |
|-----|---|
| END | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-----|---|

## address-range

Translate to an address range and port/port range.

```
address-range START END port PORT \  
port-range START END
```

### START (mandatory)

Address range start.

```
START
```

|       |                  |
|-------|------------------|
| START | An IPv4 address. |
|-------|------------------|

### END (mandatory)

Address range end.

END

END | An IPv4 address.

### port

Translate to a port.

port PORT

PORT | A 16-bit port number used by a transport protocol such as TCP or UDP.

### port-range

Translate to a port range.

port-range START END

### START (mandatory)

Port range start.

START

START | A 16-bit port number used by a transport protocol such as TCP or UDP.

### END (mandatory)

Port range end.

END

END | A 16-bit port number used by a transport protocol such as TCP or UDP.

**counters (state only)**

Counters.

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> nat source-rule <uint64> counters packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> nat source-rule <uint64> counters bytes
```

**destination-rule**

A rule to change the destination address/port of incoming packets.

```
vrouter running config# vrf <vrf> nat
vrouter running nat# destination-rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... source \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... inbound-interface [not] <string> \
... translate-to map MAP \
... address VALUE port PORT \
... port-range START END \
... address-range START END port PORT \
... port-range START END
```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### description

A comment to describe the rule.

```
description <string>
```

### protocol

Match a protocol.

```
protocol [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

The protocol to match.

```
VALUE
```

| VALUE values | Description    |
|--------------|----------------|
| tcp          | TCP protocol.  |
| udp          | UDP protocol.  |
| icmp         | ICMP protocol. |
| all          | All protocols. |

### destination

Match a destination attribute.

```
destination \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

**address**

Match this destination address or prefix.

address [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE                      | Description   |
|----------------------------|---|
| <p>&lt;domain-name&gt;</p> | <p>The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492.</p> |
| <p>&lt;A.B.C.D&gt;</p>     | <p>IPv4 address.</p>  |
| <p>&lt;A.B.C.D/M&gt;</p>   | <p>IPv4 prefix: address and CIDR mask.</p>  |

**port**

Match this destination port.

```
port [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on destination port range (syntax: port[,port|port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

### VALUE (mandatory)

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

### group

Matches a set of addresses or networks.

```
group [not] <string>
```

### not

Not match-set.

```
not
```

### <string> (mandatory)

The name of the group.

```
<string>
```

### source

Match a source attribute.

```
source \  
    address [not] VALUE \  
    port [not] VALUE \  
    port-range [not] VALUE \  
    group [not] <string>
```

**address**

Match this source address or prefix.

address [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | IPv4 prefix: address and CIDR mask.  |

**port**

Match this source port.

```
port [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

```
VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.
```

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**inbound-interface**

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

**not**

Invert the match.

```
not
```

**<string> (mandatory)**

The interface to match.

```
<string>
```

### translate-to

Translate to.

```
translate-to map MAP \  
  address VALUE port PORT \  
  port-range START END \  
  address-range START END port PORT \  
  port-range START END
```

### map

Translate a whole network of addresses onto another network of addresses. All ‘one’ bits in the mask are filled in from the new address. All bits that are zero in the mask are filled in from the original address.

```
map MAP
```

|     |  |
|-----|--|
| MAP | An IPv4 prefix: address and CIDR mask. |
|-----|--|

### address

Translate to an address and port/port range.

```
address VALUE port PORT \  
  port-range START END
```

### VALUE (mandatory)

Translate to an address.

```
VALUE
```

|       |                  |
|-------|------------------|
| VALUE | An IPv4 address. |
|-------|------------------|

### port

Translate to a port.

```
port PORT
```

|      |   |
|------|---|
| PORT | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|------|---|

### port-range

Translate to a port range.

```
port-range START END
```

### START (mandatory)

Port range start.

```
START
```

|       |   |
|-------|---|
| START | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

### END (mandatory)

Port range end.

```
END
```

|     |   |
|-----|---|
| END | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-----|---|

### address-range

Translate to an address range and port/port range.

```
address-range START END port PORT \  
port-range START END
```

### START (mandatory)

Address range start.

START

START | An IPv4 address.

### END (mandatory)

Address range end.

END

END | An IPv4 address.

### port

Translate to a port.

port PORT

PORT | A 16-bit port number used by a transport protocol such as TCP or UDP.

### port-range

Translate to a port range.

port-range START END

### START (mandatory)

Port range start.

START

START | A 16-bit port number used by a transport protocol such as TCP or UDP.

### END (mandatory)

Port range end.

```
END
```

|     |   |
|-----|---|
| END | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-----|---|

### counters (state only)

Counters.

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> nat destination-rule <uint64> counters packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> nat destination-rule <uint64> counters bytes
```

## 3.2.18 cg-nat

---

**Note:** requires a Turbo CG-NAT Application License.

---

CG-NAT configuration.

```
vrouter running config# vrf <vrf> cg-nat
```

## enabled

Enable/disable CG-NAT in this VRF.

```
vrouter running config# vrf <vrf> cg-nat
vrouter running cg-nat# enabled true|false
```

## Default value

true

## alg

Application-Level Gateway.

```
vrouter running config# vrf <vrf> cg-nat
vrouter running cg-nat# alg ALG
```

| ALG values | Description  |
|------------|--|
| ftp        | ALG for File Transfer Protocol.                              |
| h323-q931  | ALG for H.225.0 Call Signaling Protocol.                     |
| h323-ras   | ALG for H.225.0 Registration, Admission and Status Protocol. |
| pptp       | ALG for Point-to-Point Tunneling Protocol.                   |
| rtsp       | ALG for Real Time Streaming Protocol.                        |
| sip-tcp    | ALG for Session Initiation Protocol over TCP.                |
| sip-udp    | ALG for Session Initiation Protocol over UDP.                |
| tftp       | ALG for Trivial File Transfer Protocol.                      |
| dns-udp    | ALG for Domain Name System.                                  |

## pool

Pools of IP addresses for the CG-NAT rules.

```
vrouter running config# vrf <vrf> cg-nat pool <string>
```

|          |            |
|----------|------------|
| <string> | Pool name. |
|----------|------------|

## address

IPv4 addresses in the pool.

```
vrouter running config# vrf <vrf> cg-nat pool <string>
vrouter running pool <string># address ADDRESS
```

| ADDRESS values | Description                                     |
|----------------|---|
| <ipv4-address> | An IPv4 address.                                |
| <ipv4-prefix>  | An IPv4 prefix: address and CIDR mask.          |
| <ipv4-range>   | An IPv4 address range, in the form addr4-addr4. |

## block-allocation-mode

Algorithm used to associate blocks to user.

```
vrouter running config# vrf <vrf> cg-nat pool <string>
vrouter running pool <string># block-allocation-mode BLOCK-ALLOCATION-MODE
```

| BLOCK-ALLOCATION-MODE values | Description   |
|------------------------------|---|
| dynamic                      | Blocks are allocated dynamically to any user.   |
| deterministic                | Blocks are allocated deterministically. It means the same block is always allocated to the same user. |

### Default value

dynamic

## block-size

Number of ports that will be assigned to a given user.

```
vrouter running config# vrf <vrf> cg-nat pool <string>
vrouter running pool <string># block-size <uint32>
```

**port-range**

Range of ports used for each address of the pool.

```
vrouter running config# vrf <vrf> cg-nat pool <string>
vrouter running pool <string># port-range START END
```

**START**

Port range start.

```
START
```

|       |   |
|-------|---|
| START | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**END**

Port range end.

```
END
```

|     |   |
|-----|---|
| END | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-----|---|

**rule**

List of CG-NAT rules.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16>
```

|          |  |
|----------|--|
| <uint16> | Id and priority of the rule. Higher number means lower priority. |
|----------|--|

**deterministic-snat44**

Deterministic source NAT44 translation.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> deterministic-snat44
```

## match

Match parameters.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> deterministic-snat44 match
```

## outbound-interface (mandatory)

Interface to match on outbound.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> deterministic-snat44 match  
vrouter running match# outbound-interface OUTBOUND-INTERFACE
```

|                    |                    |
|--------------------|--------------------|
| OUTBOUND-INTERFACE | An interface name. |
|--------------------|--------------------|

## source

Match on source address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> deterministic-snat44 match_  
↪source
```

## ipv4-address

Match on source address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> deterministic-snat44 match_  
↪source  
vrouter running source# ipv4-address IPV4-ADDRESS
```

|              |  |
|--------------|--|
| IPV4-ADDRESS | An IPv4 prefix: address and CIDR mask. |
|--------------|--|

## translate-to

Translate to.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> deterministic-snat44 translate-  
↪to
```

### pool-name (mandatory)

Name of IP address pool used for translation.

```
vrouters running config# vrf <vrf> cg-nat rule <uint16> deterministic-snat44 translate-
↳to
vrouters running translate-to# pool-name <leafref>
```

### max-contracks-per-user

Maximum number of contracks assigned to a user. When set to 0, the number of contracks is not limited.

```
vrouters running config# vrf <vrf> cg-nat rule <uint16> deterministic-snat44 translate-
↳to
vrouters running translate-to# max-contracks-per-user <uint32>
```

### port-algo

Port allocation algorithm for new mappings.

```
vrouters running config# vrf <vrf> cg-nat rule <uint16> deterministic-snat44 translate-
↳to
vrouters running translate-to# port-algo PORT-ALGO
```

| PORT-ALGO values | Description   |
|------------------|---|
| parity           | Preserve port parity: an even port will be mapped to an even port, and an odd port will be mapped to an odd port. |
| random           | Choose port randomly.   |

### endpoint-mapping

NAT endpoint mapping behavior.

```
vrouters running config# vrf <vrf> cg-nat rule <uint16> deterministic-snat44 translate-
↳to
vrouters running translate-to# endpoint-mapping ENDPOINT-MAPPING
```

| ENDPOINT-MAPPING values | Description   |
|-------------------------|---|
| dependent               | Reuse port mapping for subsequent packets sent from the same internal IP address and port to the same external IP address and port. |
| independent             | Reuse the port mapping for subsequent packets sent from the same internal IP address and port to any external IP address and port.  |

## endpoint-filtering

NAT endpoint filtering behavior.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> deterministic-snat44 translate-
↳to
vrouter running translate-to# endpoint-filtering ENDPOINT-FILTERING
```

| ENDPOINT-FILTERING values | Description  |
|---------------------------|--|
| dependent                 | Inbound packets from external endpoints are filtered out if they don't fully match an existing mapping (IP/port src/dst).  |
| independent               | Inbound packets from external endpoints are filtered out only if their destination IP address and port don't match an existing mapping (IP/port src can differ). |

## hairpinning

Enable communication between two hosts on the internal network, using their mapped endpoint.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> deterministic-snat44 translate-
↳to
vrouter running translate-to# hairpinning true|false
```

## address-pooling

CG-NAT Address Pooling mode.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> deterministic-snat44 translate-
↳to
vrouter running translate-to# address-pooling ADDRESS-POOLING
```

| ADDRESS-POOLING values | Description   |
|------------------------|---|
| paired                 | In paired mode, the same IP of the pool is used to translate all the sessions originating from the same CPE.            |
| no-paired              | In no-paired mode, different IPs of the pool can be used to translate different sessions originating from the same CPE. |

### dynamic-snat44

Dynamic source NAT44 translation.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44
```

### match

Match parameters.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 match
```

### outbound-interface (mandatory)

Interface to match on outbound.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 match
vrouter running match# outbound-interface OUTBOUND-INTERFACE
```

|                    |                    |
|--------------------|--------------------|
| OUTBOUND-INTERFACE | An interface name. |
|--------------------|--------------------|

### source

Match on source address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 match source
```

## ipv4-address

Match on source address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 match source
vrouter running source# ipv4-address IPV4-ADDRESS
```

|              |  |
|--------------|--|
| IPV4-ADDRESS | An IPv4 prefix: address and CIDR mask. |
|--------------|--|

## translate-to

Translate to.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 translate-to
```

## pool-name (mandatory)

Name of IP address pool used for translation.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 translate-to
vrouter running translate-to# pool-name <leafref>
```

## max-blocks-per-user

Maximum number of port blocks assigned to a user.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 translate-to
vrouter running translate-to# max-blocks-per-user <uint16>
```

## active-block-timeout

Interval during which the the current block is used to allocate sessions. When set to 0, the current block is always used.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 translate-to
vrouter running translate-to# active-block-timeout <uint16>
```

## user-timeout

Interval during which the current block remains active after all user flows have expired.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 translate-to
vrouter running translate-to# user-timeout <uint16>
```

## max-contracks-per-user

Maximum number of contracks assigned to a user. When set to 0, the number of contracks is not limited.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 translate-to
vrouter running translate-to# max-contracks-per-user <uint32>
```

## port-algo

Port allocation algorithm for new mappings.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 translate-to
vrouter running translate-to# port-algo PORT-ALGO
```

| PORT-ALGO values | Description   |
|------------------|---|
| parity           | Preserve port parity: an even port will be mapped to an even port, and an odd port will be mapped to an odd port. |
| random           | Choose port randomly.   |

## endpoint-mapping

NAT endpoint mapping behavior.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 translate-to
vrouter running translate-to# endpoint-mapping ENDPOINT-MAPPING
```

| ENDPOINT-MAPPING values | Description   |
|-------------------------|---|
| dependent               | Reuse port mapping for subsequent packets sent from the same internal IP address and port to the same external IP address and port. |
| independent             | Reuse the port mapping for subsequent packets sent from the same internal IP address and port to any external IP address and port.  |

## endpoint-filtering

NAT endpoint filtering behavior.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 translate-to
vrouter running translate-to# endpoint-filtering ENDPOINT-FILTERING
```

| ENDPOINT-FILTERING values | Description  |
|---------------------------|--|
| dependent                 | Inbound packets from external endpoints are filtered out if they don't fully match an existing mapping (IP/port src/dst).  |
| independent               | Inbound packets from external endpoints are filtered out only if their destination IP address and port don't match an existing mapping (IP/port src can differ). |

## hairpinning

Enable communication between two hosts on the internal network, using their mapped endpoint.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 translate-to
vrouter running translate-to# hairpinning true|false
```

## address-pooling

CG-NAT Address Pooling mode.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat44 translate-to
vrouter running translate-to# address-pooling ADDRESS-POOLING
```

| ADDRESS-POOLING values | Description   |
|------------------------|---|
| paired                 | In paired mode, the same IP of the pool is used to translate all the sessions originating from the same CPE.            |
| no-paired              | In no-paired mode, different IPs of the pool can be used to translate different sessions originating from the same CPE. |

## dynamic-snat64

Dynamic source NAT64 translation.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64
```

## match

Match parameters.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 match
```

## outbound-interface (mandatory)

Interface to match on outbound.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 match  
vrouter running match# outbound-interface OUTBOUND-INTERFACE
```

|                    |                    |
|--------------------|--------------------|
| OUTBOUND-INTERFACE | An interface name. |
|--------------------|--------------------|

## source

Match on source address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 match source
```

## ipv6-address

Match on source address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 match source  
vrouter running source# ipv6-address IPV6-ADDRESS
```

|              |  |
|--------------|--|
| IPV6-ADDRESS | An IPv6 prefix: address and CIDR mask. |
|--------------|--|

## translate-to

Translate to.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 translate-to
```

## pool-name (mandatory)

Name of IP address pool used for translation.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 translate-to  
vrouter running translate-to# pool-name <leafref>
```

## max-blocks-per-user

Maximum number of port blocks assigned to a user.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 translate-to  
vrouter running translate-to# max-blocks-per-user <uint16>
```

## active-block-timeout

Interval during which the the current block is used to allocate sessions. When set to 0, the current block is always used.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 translate-to  
vrouter running translate-to# active-block-timeout <uint16>
```

## user-timeout

Interval during which the current block remains active after all user flows have expired.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 translate-to  
vrouter running translate-to# user-timeout <uint16>
```

## max-contracks-per-user

Maximum number of contracks assigned to a user. When set to 0, the number of contracks is not limited.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 translate-to
vrouter running translate-to# max-contracks-per-user <uint32>
```

## port-algo

Port allocation algorithm for new mappings.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 translate-to
vrouter running translate-to# port-algo PORT-ALGO
```

| PORT-ALGO values | Description   |
|------------------|---|
| parity           | Preserve port parity: an even port will be mapped to an even port, and an odd port will be mapped to an odd port. |
| random           | Choose port randomly.   |

## endpoint-mapping

NAT endpoint mapping behavior.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 translate-to
vrouter running translate-to# endpoint-mapping ENDPOINT-MAPPING
```

| ENDPOINT-MAPPING values | Description   |
|-------------------------|---|
| dependent               | Reuse port mapping for subsequent packets sent from the same internal IP address and port to the same external IP address and port. |
| independent             | Reuse the port mapping for subsequent packets sent from the same internal IP address and port to any external IP address and port.  |

## endpoint-filtering

NAT endpoint filtering behavior.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 translate-to
vrouter running translate-to# endpoint-filtering ENDPOINT-FILTERING
```

| ENDPOINT-FILTERING values | Description  |
|---------------------------|--|
| dependent                 | Inbound packets from external endpoints are filtered out if they don't fully match an existing mapping (IP/port src/dst).  |
| independent               | Inbound packets from external endpoints are filtered out only if their destination IP address and port don't match an existing mapping (IP/port src can differ). |

## hairpinning

Enable communication between two hosts on the internal network, using their mapped endpoint.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 translate-to
vrouter running translate-to# hairpinning true|false
```

## address-pooling

CG-NAT Address Pooling mode.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 translate-to
vrouter running translate-to# address-pooling ADDRESS-POOLING
```

| ADDRESS-POOLING values | Description   |
|------------------------|---|
| paired                 | In paired mode, the same IP of the pool is used to translate all the sessions originating from the same CPE.            |
| no-paired              | In no-paired mode, different IPs of the pool can be used to translate different sessions originating from the same CPE. |

## destination-prefix

NAT64 destination prefix.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> dynamic-snat64 translate-to  
vrouter running translate-to# destination-prefix DESTINATION-PREFIX
```

|                    |  |
|--------------------|--|
| DESTINATION-PREFIX | An IPv6 prefix: address and CIDR mask. |
|--------------------|--|

## static-dnat44

Static destination NAT44 translation.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat44
```

## match

Match parameters.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat44 match
```

## inbound-interface (mandatory)

Interface to match on inbound.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat44 match  
vrouter running match# inbound-interface INBOUND-INTERFACE
```

|                   |                    |
|-------------------|--------------------|
| INBOUND-INTERFACE | An interface name. |
|-------------------|--------------------|

## destination

Match on destination address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat44 match destination
```

## ipv4-address

Match on destination address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat44 match destination
vrouter running destination# ipv4-address IPV4-ADDRESS
```

|              |  |
|--------------|--|
| IPV4-ADDRESS | An IPv4 prefix: address and CIDR mask. |
|--------------|--|

## translate-to

Translate to.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat44 translate-to
```

## ipv4-address (mandatory)

Translated Address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat44 translate-to
vrouter running translate-to# ipv4-address IPV4-ADDRESS
```

|              |                  |
|--------------|------------------|
| IPV4-ADDRESS | An IPv4 address. |
|--------------|------------------|

## static-dnat46

Static destination NAT46 translation.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat46
```

## match

Match parameters.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat46 match
```

### inbound-interface (mandatory)

Interface to match on inbound.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat46 match
vrouter running match# inbound-interface INBOUND-INTERFACE
```

|                   |                    |
|-------------------|--------------------|
| INBOUND-INTERFACE | An interface name. |
|-------------------|--------------------|

### destination

Match on destination address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat46 match destination
```

### ipv4-address

Match on destination address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat46 match destination
vrouter running destination# ipv4-address IPV4-ADDRESS
```

|              |  |
|--------------|--|
| IPV4-ADDRESS | An IPv4 prefix: address and CIDR mask. |
|--------------|--|

### translate-to

Translate to.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat46 translate-to
```

### ipv6-address (mandatory)

Translated Address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat46 translate-to
vrouter running translate-to# ipv6-address IPV6-ADDRESS
```

|              |                  |
|--------------|------------------|
| IPV6-ADDRESS | An IPv6 address. |
|--------------|------------------|

## source-prefix

NAT46 source prefix.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-dnat46 translate-to  
vrouter running translate-to# source-prefix SOURCE-PREFIX
```

|               |  |
|---------------|--|
| SOURCE-PREFIX | An IPv6 prefix: address and CIDR mask. |
|---------------|--|

## static-snat44

Static source NAT44 translation.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat44
```

## match

Match parameters.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat44 match
```

## outbound-interface (mandatory)

Interface to match on outbound.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat44 match  
vrouter running match# outbound-interface OUTBOUND-INTERFACE
```

|                    |                    |
|--------------------|--------------------|
| OUTBOUND-INTERFACE | An interface name. |
|--------------------|--------------------|

## source

Match on source address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat44 match source
```

## ipv4-address

Match on source address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat44 match source  
vrouter running source# ipv4-address IPV4-ADDRESS
```

|              |  |
|--------------|--|
| IPV4-ADDRESS | An IPv4 prefix: address and CIDR mask. |
|--------------|--|

## translate-to

Translate to.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat44 translate-to
```

## ipv4-address (mandatory)

Translated Address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat44 translate-to  
vrouter running translate-to# ipv4-address IPV4-ADDRESS
```

|              |                  |
|--------------|------------------|
| IPV4-ADDRESS | An IPv4 address. |
|--------------|------------------|

## static-snat64

Static source NAT64 translation.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat64
```

## match

Match parameters.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat64 match
```

**outbound-interface (mandatory)**

Interface to match on outbound.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat64 match
vrouter running match# outbound-interface OUTBOUND-INTERFACE
```

|                    |                    |
|--------------------|--------------------|
| OUTBOUND-INTERFACE | An interface name. |
|--------------------|--------------------|

**source**

Match on source address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat64 match source
```

**ipv6-address**

Match on source address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat64 match source
vrouter running source# ipv6-address IPV6-ADDRESS
```

|              |  |
|--------------|--|
| IPV6-ADDRESS | An IPv6 prefix: address and CIDR mask. |
|--------------|--|

**translate-to**

Translate to.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat64 translate-to
```

**ipv4-address (mandatory)**

Translated Address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat64 translate-to
vrouter running translate-to# ipv4-address IPV4-ADDRESS
```

|              |                  |
|--------------|------------------|
| IPV4-ADDRESS | An IPv4 address. |
|--------------|------------------|

## destination-prefix

NAT64 destination prefix.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> static-snat64 translate-to
vrouter running translate-to# destination-prefix DESTINATION-PREFIX
```

|                    |  |
|--------------------|--|
| DESTINATION-PREFIX | An IPv6 prefix: address and CIDR mask. |
|--------------------|--|

## match

Match parameters.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> match
```

## outbound-interface (mandatory)

Interface to match on outbound.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> match
vrouter running match# outbound-interface OUTBOUND-INTERFACE
```

|                    |                    |
|--------------------|--------------------|
| OUTBOUND-INTERFACE | An interface name. |
|--------------------|--------------------|

## source

Match on source address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> match source
```

## address

Match on source address.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> match source
vrouter running source# address ADDRESS
```

|         |  |
|---------|--|
| ADDRESS | An IPv4 prefix: address and CIDR mask. |
|---------|--|

## translate-to

Translate to.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> translate-to
```

## pool-name (mandatory)

Name of IP address pool used for translation.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> translate-to  
vrouter running translate-to# pool-name <leafref>
```

## max-blocks-per-user

Maximum number of port blocks assigned to a user.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> translate-to  
vrouter running translate-to# max-blocks-per-user <uint16>
```

## active-block-timeout

Interval during which the the current block is used to allocate sessions. When set to 0, the current block is always used.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> translate-to  
vrouter running translate-to# active-block-timeout <uint16>
```

## user-timeout

Interval during which the current block remains active after all user flows have expired.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> translate-to  
vrouter running translate-to# user-timeout <uint16>
```

### max-contracks-per-user

Maximum number of contracks assigned to a user. When set to 0, the number of contracks is not limited.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> translate-to
vrouter running translate-to# max-contracks-per-user <uint32>
```

### port-algo

Port allocation algorithm for new mappings.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> translate-to
vrouter running translate-to# port-algo PORT-ALGO
```

| PORT-ALGO values | Description   |
|------------------|---|
| parity           | Preserve port parity: an even port will be mapped to an even port, and an odd port will be mapped to an odd port. |
| random           | Choose port randomly.   |

### endpoint-mapping

NAT endpoint mapping behavior.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> translate-to
vrouter running translate-to# endpoint-mapping ENDPOINT-MAPPING
```

| ENDPOINT-MAPPING values | Description   |
|-------------------------|---|
| dependent               | Reuse port mapping for subsequent packets sent from the same internal IP address and port to the same external IP address and port. |
| independent             | Reuse the port mapping for subsequent packets sent from the same internal IP address and port to any external IP address and port.  |

## endpoint-filtering

NAT endpoint filtering behavior.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> translate-to
vrouter running translate-to# endpoint-filtering ENDPOINT-FILTERING
```

| ENDPOINT-FILTERING values | Description  |
|---------------------------|--|
| dependent                 | Inbound packets from external endpoints are filtered out if they don't fully match an existing mapping (IP/port src/dst).  |
| independent               | Inbound packets from external endpoints are filtered out only if their destination IP address and port don't match an existing mapping (IP/port src can differ). |

## hairpinning

Enable communication between two hosts on the internal network, using their mapped endpoint.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> translate-to
vrouter running translate-to# hairpinning true|false
```

## address-pooling

CG-NAT Address Pooling mode.

```
vrouter running config# vrf <vrf> cg-nat rule <uint16> translate-to
vrouter running translate-to# address-pooling ADDRESS-POOLING
```

| ADDRESS-POOLING values | Description   |
|------------------------|---|
| paired                 | In paired mode, the same IP of the pool is used to translate all the sessions originating from the same CPE.            |
| no-paired              | In no-paired mode, different IPs of the pool can be used to translate different sessions originating from the same CPE. |

## contrack

Contrack options.

```
vrouter running config# vrf <vrf> cg-nat contrack
```

## behavior

Specific TCP options.

```
vrouter running config# vrf <vrf> cg-nat contrack
vrouter running contrack# behavior <behavior> enabled true|false
```

| <behavior> values    | Description           |
|----------------------|-----------------------|
| tcp-window-check     | TCP window check.     |
| tcp-rst-strict-order | TCP rst strict order. |

## enabled (mandatory)

Enable option.

```
enabled true|false
```

## timeouts

Timeouts for the different events/protocols.

```
vrouter running config# vrf <vrf> cg-nat contrack timeouts
```

## icmp

Contrack options for ICMP.

```
vrouter running config# vrf <vrf> cg-nat contrack timeouts
vrouter running timeouts# icmp <icmp> <uint32>
```

| <icmp> values | Description        |
|---------------|--------------------|
| new           | State NEW.         |
| established   | State ESTABLISHED. |
| closed        | State CLOSED.      |

**<uint32> (mandatory)**

Timeout in seconds.

```
<uint32>
```

**udp**

Conntrack options for UDP.

```
vrouter running config# vrf <vrf> cg-nat conntrack timeouts
vrouter running timeouts# udp <udp> <uint32>
```

| <udp> values | Description        |
|--------------|--------------------|
| new          | State NEW.         |
| established  | State ESTABLISHED. |
| closed       | State CLOSED.      |

**<uint32> (mandatory)**

Timeout in seconds.

```
<uint32>
```

**gre-pptp**

Conntrack options for GRE-PPTP.

```
vrouter running config# vrf <vrf> cg-nat conntrack timeouts
vrouter running timeouts# gre-pptp <gre-pptp> <uint32>
```

| <gre-pptp> values | Description        |
|-------------------|--------------------|
| new               | State NEW.         |
| established       | State ESTABLISHED. |
| closed            | State CLOSED.      |

**<uint32> (mandatory)**

Timeout in seconds.

```
<uint32>
```

**tcp**

Conntrack options for TCP.

```
vrouter running config# vrf <vrf> cg-nat conntrack timeouts  
vrouter running timeouts# tcp <tcp> <uint32>
```

| <tcp> values | Description         |
|--------------|---------------------|
| syn-sent     | State SYN-SENT.     |
| simsyn-sent  | State SIMSYN-SENT.  |
| syn-received | State SYN-RECEIVED. |
| established  | State ESTABLISHED.  |
| fin-sent     | State FIN-SENT.     |
| fin-received | State FIN-RECEIVED. |
| closed       | State CLOSED.       |
| close-wait   | State CLOSE-WAIT.   |
| fin-wait     | State FIN-WAIT.     |
| last-ack     | State LAST-ACK.     |
| time-wait    | State TIME-WAIT.    |

**<uint32> (mandatory)**

Timeout in seconds.

```
<uint32>
```

**nat64**

NAT64 conntrack options.

```
vrouter running config# vrf <vrf> cg-nat conntrack nat64
```

**option**

Specific NAT64 options.

```
vrouter running config# vrf <vrf> cg-nat conntrack nat64
vrouter running nat64# option <option> true|false
```

| <option> values        | Description   |
|------------------------|---|
| update-tcp-mss         | Enable/Disable TCP MSS update.                                |
| drop-udp-zero-checksum | Enable/Disable UDP null checksum packet drops.                |
| force-frag-ipv4        | Fragment IPv4 packets (with DF flag) if the MTU is too small. |
| force-frag-ipv6        | Fragment IPv6 packets if the MTU is too small.                |

**true|false (mandatory)**

Option state.

```
true|false
```

**mtu**

NAT64 lowest IPv6 mtu configuration.

```
vrouter running config# vrf <vrf> cg-nat conntrack nat64
vrouter running nat64# mtu <mtu> <uint16>
```

|       |                      |
|-------|----------------------|
| <mtu> | Set lowest IPv6 MTU. |
|-------|----------------------|

**<uint16> (mandatory)**

MTU (0 to fragment packet according to the MTU of the output interface).

```
<uint16>
```

## logging

CG-NAT log configuration.

```
vrouter running config# vrf <vrf> cg-nat logging
```

### enabled

Enable log.

```
vrouter running config# vrf <vrf> cg-nat logging
vrouter running logging# enabled true|false
```

## 3.2.19 ntp

Top-level container for NTP configuration.

```
vrouter running config# vrf <vrf> ntp
```

### enabled

Enable or disable the NTP protocol and indicates that the system should attempt to synchronize the system clock with an NTP server from the servers defined in the 'ntp/server' list.

```
vrouter running config# vrf <vrf> ntp
vrouter running ntp# enabled true|false
```

### Default value

true

### ntp-source-address

Source address to use on outgoing NTP packets.

```
vrouter running config# vrf <vrf> ntp
vrouter running ntp# ntp-source-address NTP-SOURCE-ADDRESS
```

| NTP-SOURCE-ADDRESS values | Description      |
|---------------------------|------------------|
| <A.B.C.D>                 | An IPv4 address. |
| <X:X::X:X>                | An IPv6 address. |

### auth-key

List of NTP authentication keys.

```
vrouter running config# vrf <vrf> ntp auth-key <uint16>
```

|                             |   |
|-----------------------------|---|
| <code>&lt;uint16&gt;</code> | Integer identifier used by the client and server to designate a secret key. The client and server must use the same key id. |
|-----------------------------|---|

### key-value

NTP authentication key value.

```
vrouter running config# vrf <vrf> ntp auth-key <uint16>
vrouter running auth-key <uint16># key-value <string>
```

### server

List of NTP servers to use for system clock synchronization. If '/system/ntp/enabled' is 'true', then the system will attempt to contact and utilize the specified NTP servers.

```
vrouter running config# vrf <vrf> ntp server <server>
```

| <code>&lt;server values&gt;</code> | Description  |
|------------------------------------|--|
| <code>&lt;A.B.C.D&gt;</code>       | IPv4 address.  |
| <code>&lt;X:X::X:X&gt;</code>      | IPv6 address.  |
| <code>&lt;host-name&gt;</code>     | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |

**version**

Version number to put in outgoing NTP packets.

```
vrouter running config# vrf <vrf> ntp server <server>
vrouter running server <server># version <uint8>
```

**Default value**

4

**association-type**

The desired association type for this NTP server.

```
vrouter running config# vrf <vrf> ntp server <server>
vrouter running server <server># association-type ASSOCIATION-TYPE
```

| ASSOCIATION-TYPE values | Description  |
|-------------------------|--|
| SERVER                  | Use client association mode. This device will not provide synchronization to the configured NTP server.  |
| PEER                    | Use symmetric active association mode. This device may provide synchronization to the configured NTP server.   |
| POOL                    | Use client association mode with one or more of the NTP servers found by DNS resolution of the domain name given by the 'address' leaf. This device will not provide synchronization to the servers. |

**Default value**

SERVER

**iburst**

Indicates whether this server should enable burst synchronization or not.

```
vrouter running config# vrf <vrf> ntp server <server>
vrouter running server <server># iburst true|false
```

**Default value**

false

## prefer

Indicates whether this server should be preferred or not.

```
vrouter running config# vrf <vrf> ntp server <server>  
vrouter running server <server># prefer true|false
```

### Default value

false

## auth-key-id

Integer identifier used by the client and server to designate a secret key. The client and server must use the same key id.

```
vrouter running config# vrf <vrf> ntp server <server>  
vrouter running server <server># auth-key-id <leafref>
```

## stratum (state only)

Indicates the level of the server in the NTP hierarchy. As stratum number increases, the accuracy is degraded. Primary servers are stratum while a maximum value of 16 indicates unsynchronized. The values have the following specific semantics: | 0 | unspecified or invalid | 1 | primary server (e.g., equipped with a GPS receiver) | 2-15 | secondary server (via NTP) | 16 | unsynchronized | 17-255 | reserved.

```
vrouter> show state vrf <vrf> ntp server <server> stratum
```

## root-delay (state only)

The round-trip delay to the server, in milliseconds.

```
vrouter> show state vrf <vrf> ntp server <server> root-delay
```

## root-dispersion (state only)

Dispersion (epsilon) represents the maximum error inherent in the measurement.

```
vrouter> show state vrf <vrf> ntp server <server> root-dispersion
```

### offset (state only)

Estimate of the current time offset from the peer. This is the time difference between the local and reference clock.

```
vrouter> show state vrf <vrf> ntp server <server> offset
```

### poll-interval (state only)

Polling interval of the peer.

```
vrouter> show state vrf <vrf> ntp server <server> poll-interval
```

### synchronized (state only)

True if we are synchronized with this server.

```
vrouter> show state vrf <vrf> ntp server <server> synchronized
```

### state (state only)

The server status in the clock selection process.

```
vrouter> show state vrf <vrf> ntp server <server> state
```

## 3.2.20 firewall

### ipv4 filter

---

**Note:** requires a Turbo Router Network License.

---

Default table.

```
vrouter running config# vrf <vrf> firewall ipv4 filter
```

## input

Packets destined to local sockets.

```
vrouter running config# vrf <vrf> firewall ipv4 filter input
```

## policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv4 filter input
vrouter running input# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## Default value

accept

## packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 filter input packets
```

## bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 filter input bytes
```

## rule

A rule to perform an action on matching packets.

```

vrouters running config# vrf <vrf> firewall ipv4 filter input
vrouters running input# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... source \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... ipv4 [not] fragment \
... icmp-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
...   status [not] VALUE \
...   state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
...   rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... inbound-interface [not] <string> \
... action STANDARD chain <leafref> reject REJECT \
...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### **description**

A comment to describe the rule.

```
description <string>
```

### **protocol**

Match the protocol.

```
protocol [not] VALUE
```

### **not**

Invert the match.

```
not
```

### **VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| icmp         | ICMP protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

## destination

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

## address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | prefix: address and CIDR mask.   |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C/DIM>   | prefix: address and CIDR mask.   |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**ipv4**

Match the fragment.

```
ipv4 [not] fragment
```

**not**

Invert the match.

```
not
```

**fragment (mandatory)**

Match if the packet is a fragment.

```
fragment
```

### icmp-type

Match the packet ICMP type.

icmp-type [not] VALUE

### not

Invert the match.

not

### VALUE (mandatory)

The ICMP type to match.

VALUE

| VALUE values              | Description                |
|---------------------------|----------------------------|
| any                       | Any ICMP type.             |
| echo-request              | Echo request.              |
| echo-reply                | Echo reply.                |
| destination-unreachable   | Destination unreachable.   |
| network-unreachable       | Network unreachable.       |
| host-unreachable          | Host unreachable.          |
| protocol-unreachable      | Protocol unreachable.      |
| port-unreachable          | Port unreachable.          |
| fragmentation-needed      | Fragmentation needed.      |
| source-route-failed       | Source route failed.       |
| network-unknown           | Network unknown.           |
| host-unknown              | Host unknown.              |
| network-prohibited        | Network prohibited.        |
| host-prohibited           | Host prohibited.           |
| TOS-network-unreachable   | TOS network unreachable.   |
| TOS-host-unreachable      | TOS host unreachable.      |
| communication-prohibited  | Communication prohibited.  |
| host-precedence-violation | Host precedence violation. |
| precedence-cutoff         | Precedence cutoff.         |
| source-quench             | Source quench.             |
| redirect                  | Redirect.                  |
| network-redirect          | Network redirect.          |

continues on next page

Table 2 – continued from previous page

| VALUE values               | Description                        |
|----------------------------|------------------------------------|
| host-redirect              | Host redirect.                     |
| TOS-network-redirect       | TOS network redirect.              |
| TOS-host-redirect          | TOS host redirect.                 |
| router-advertisement       | Router advertisement.              |
| router-solicitation        | Router solicitation.               |
| ttl-exceeded               | TTL exceeded.                      |
| ttl-zero-during-transit    | Time to Live exceeded in Transit.  |
| ttl-zero-during-reassembly | Fragment Reassembly Time Exceeded. |
| parameter-problem          | Parameter problem.                 |
| ip-header-bad              | Bad IP header.                     |
| required-option-missing    | Missing a Required Option.         |
| timestamp-request          | Timestamp request.                 |
| timestamp-reply            | Timestamp reply.                   |
| address-mask-request       | Address mask request.              |
| address-mask-reply         | Address mask reply.                |

**tcp-flags**

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

**not**

Invert the match.

```
not
```

**set**

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

## examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

## contrack

Match contrack information.

```
contrack \  
  status [not] VALUE \  
  state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \
  shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \
  forward-tsn \
  data examined EXAMINED set SET \
  abort examined EXAMINED set SET \
  shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

```
abort examined EXAMINED set SET
```

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

```
shutdown-complete examined EXAMINED set SET
```

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

## action

The action performed by this rule.

```

action STANDARD chain <leafref> reject REJECT \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

## STANDARD

Standard action.

|          |
|----------|
| STANDARD |
|----------|

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## chain

Jump to the user chain by this name.

|                 |
|-----------------|
| chain <leafref> |
|-----------------|

## reject

Used to send back an error packet in response to the matched packet.

|               |
|---------------|
| reject REJECT |
|---------------|

| REJECT values          | Description   |
|------------------------|---|
| icmp-net-unreachable   | Reject with ICMP network unreachable.   |
| icmp-host-unreachable  | Reject with ICMP host unreachable.  |
| icmp-port-unreachable  | Reject with ICMP port unreachable.  |
| icmp-proto-unreachable | Reject with ICMP prototype unreachable.   |
| icmp-net-prohibited    | Reject with ICMP network prohibited.  |
| icmp-host-prohibited   | Reject with ICMP host prohibited.   |
| icmp-admin-prohibited  | Reject with ICMP admin prohibited.  |
| tcp-reset              | Reject with TCP RST packet. Can be used on rules which only match the TCP protocol. |

### connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

### <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

### mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nfmak) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmak \& nfmask)$  i.e. ctmask defines what bits to clear and nfmak what bits of the nfmak to XOR into the ctmark. ctmask and nfmak default to 0xFFFFFFFF.

```
save-mark nfmak <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmak

Bits that should be XORed into the connection mark.

```
nfmak <0x0-0xffffffff>
```

## ctmask

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

## restore-mark

Copy the connection mark (ctmark) to the packet mark (nfmak) using the given masks. The new ctmark value is determined as follows:  $nfmak = (nfmak \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmak defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmak default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmak <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmak

Bits that should be cleared.

```
nfmak <0x0-0xffffffff>
```

### ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

### prefix

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

**tcpmss**

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

### clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

### counters (state only)

The counters of this rule.

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 filter input rule <uint64> counters packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 filter input rule <uint64> counters bytes
```

### forward

Packets being routed.

```
vrouter running config# vrf <vrf> firewall ipv4 filter forward
```

## policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv4 filter forward
vrouter running forward# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### Default value

accept

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 filter forward packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 filter forward bytes
```

## rule

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv4 filter forward
vrouter running forward# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
```

(continues on next page)

(continued from previous page)

```

... source \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... ipv4 [not] fragment \
... icmp-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
...   status [not] VALUE \
...   state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
...   rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... inbound-interface [not] <string> \
... outbound-interface [not] <string> \
... action STANDARD chain <leafref> reject REJECT \
...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

**description**

A comment to describe the rule.

```
description <string>
```

**protocol**

Match the protocol.

```
protocol [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| icmp         | ICMP protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

## destination

Match on destination fields.

```
destination \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on destination address.

```
address [not] VALUE
```

## not

Invert the match.

```
not
```

## VALUE (mandatory)

The address to match.

```
VALUE
```

| VALUE         | Description  |
|---------------|--|
| val-ues       |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | prefix: address and CIDR mask.   |

### port

Match on destination port.

port [not] VALUE

### not

Invert the match.

not

### VALUE (mandatory)

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

## port-range

Match on destination port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

### not

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**source**

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

**address**

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The address to match.

```
VALUE
```

| VALUE       | Description  |
|-------------|--|
| val-ues     |  |
| <domain>    | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>   | IPv4 address.  |
| <A.B.C.D/M> | prefix: address and CIDR mask.   |

### port

Match on source port.

port [not] VALUE

### not

Invert the match.

not

### VALUE (mandatory)

The port to match.

VALUE

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

## port-range

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

## not

Invert the match.

```
not
```

## VALUE (mandatory)

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**ipv4**

Match the fragment.

```
ipv4 [not] fragment
```

**not**

Invert the match.

```
not
```

**fragment (mandatory)**

Match if the packet is a fragment.

```
fragment
```

**icmp-type**

Match the packet ICMP type.

```
icmp-type [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The ICMP type to match.

|       |
|-------|
| VALUE |
|-------|

| VALUE values               | Description                        |
|----------------------------|------------------------------------|
| any                        | Any ICMP type.                     |
| echo-request               | Echo request.                      |
| echo-reply                 | Echo reply.                        |
| destination-unreachable    | Destination unreachable.           |
| network-unreachable        | Network unreachable.               |
| host-unreachable           | Host unreachable.                  |
| protocol-unreachable       | Protocol unreachable.              |
| port-unreachable           | Port unreachable.                  |
| fragmentation-needed       | Fragmentation needed.              |
| source-route-failed        | Source route failed.               |
| network-unknown            | Network unknown.                   |
| host-unknown               | Host unknown.                      |
| network-prohibited         | Network prohibited.                |
| host-prohibited            | Host prohibited.                   |
| TOS-network-unreachable    | TOS network unreachable.           |
| TOS-host-unreachable       | TOS host unreachable.              |
| communication-prohibited   | Communication prohibited.          |
| host-precedence-violation  | Host precedence violation.         |
| precedence-cutoff          | Precedence cutoff.                 |
| source-quench              | Source quench.                     |
| redirect                   | Redirect.                          |
| network-redirect           | Network redirect.                  |
| host-redirect              | Host redirect.                     |
| TOS-network-redirect       | TOS network redirect.              |
| TOS-host-redirect          | TOS host redirect.                 |
| router-advertisement       | Router advertisement.              |
| router-solicitation        | Router solicitation.               |
| ttl-exceeded               | TTL exceeded.                      |
| ttl-zero-during-transit    | Time to Live exceeded in Transit.  |
| ttl-zero-during-reassembly | Fragment Reassembly Time Exceeded. |
| parameter-problem          | Parameter problem.                 |
| ip-header-bad              | Bad IP header.                     |
| required-option-missing    | Missing a Required Option.         |
| timestamp-request          | Timestamp request.                 |
| timestamp-reply            | Timestamp reply.                   |
| address-mask-request       | Address mask request.              |

continues on next page

Table 3 – continued from previous page

| VALUE values       | Description         |
|--------------------|---------------------|
| address-mask-reply | Address mask reply. |

**tcp-flags**

Match the packet TCP flags.

tcp-flags [not] set SET examined EXAMINED

**not**

Invert the match.

not

**set**

Set flags.

set SET

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

**examined**

Examined flags.

examined EXAMINED

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

### status

Match the connection status.

```
status [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE

The contrack status to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

### state

Match the packet state regarding conntrack.

```
state [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE

The packet states to match.

```
VALUE
```

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

**connmark**

Matches the mark field associated with a connection.

```
connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

```
dscp [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The DSCP value to match.

|       |
|-------|
| VALUE |
|-------|

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

|                                      |
|--------------------------------------|
| tos [not] <0x0-0xff> mask <0x0-0xff> |
|--------------------------------------|

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**sctp-chunk-types**

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \
  shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \
  forward-tsn \
  data examined EXAMINED set SET \
  abort examined EXAMINED set SET \
  shutdown-complete examined EXAMINED set SET
```

**not**

Invert the match.

```
not
```

**SCOPE (mandatory)**

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

**init**

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

set SET

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

**abort**

ABORT chunk.

abort examined EXAMINED set SET

**examined**

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

**set**

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

```
shutdown-complete examined EXAMINED set SET
```

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

**<string> (mandatory)**

The interface to match.

```
<string>
```

**outbound-interface**

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

**not**

Invert the match.

```
not
```

**<string> (mandatory)**

The interface to match.

```
<string>
```

**action**

The action performed by this rule.

```
action STANDARD chain <leafref> reject REJECT \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

## STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## chain

Jump to the user chain by this name.

```
chain <leafref>
```

## reject

Used to send back an error packet in response to the matched packet.

```
reject REJECT
```

| REJECT values          | Description   |
|------------------------|---|
| icmp-net-unreachable   | Reject with ICMP network unreachable.   |
| icmp-host-unreachable  | Reject with ICMP host unreachable.  |
| icmp-port-unreachable  | Reject with ICMP port unreachable.  |
| icmp-proto-unreachable | Reject with ICMP prototype unreachable.   |
| icmp-net-prohibited    | Reject with ICMP network prohibited.  |
| icmp-host-prohibited   | Reject with ICMP host prohibited.   |
| icmp-admin-prohibited  | Reject with ICMP admin prohibited.  |
| tcp-reset              | Reject with TCP RST packet. Can be used on rules which only match the TCP protocol. |

**connmark**

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \  
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \  
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \  
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

**set-xmark**

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

XOR with this value.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

**save-mark**

Copy the packet mark (nmask) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nmask \& nmask)$  i.e. ctmask defines what bits to clear and nmask what bits of the nmask to XOR into the ctmark. ctmask and nmask default to 0xFFFFFFFF.

```
save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### **nfmask**

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

### **ctmask**

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

### **restore-mark**

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $\text{nfmark} = (\text{nfmark} \& \sim\text{nfmask}) \wedge (\text{ctmark} \& \text{ctmask})$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### **nfmask**

Bits that should be cleared.

```
nfmask <0x0-0xffffffff>
```

### **ctmask**

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

## log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

## level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

## prefix

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

## additional-infos

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

## mark

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

### <0x0-0xffffffff> (mandatory)

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

### clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

**counters (state only)**

The counters of this rule.

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 filter forward rule <uint64> counters_
↳packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 filter forward rule <uint64> counters bytes
```

**output**

Locally-generated packets.

```
vrouter running config# vrf <vrf> firewall ipv4 filter output
```

**policy**

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv4 filter output
vrouter running output# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

**Default value**

accept

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 filter output packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 filter output bytes
```

**rule**

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv4 filter output
vrouter running output# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... source \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... ipv4 [not] fragment \
... icmp-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
... status [not] VALUE \
... state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
... rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
```

(continues on next page)

(continued from previous page)

```

... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
... data examined EXAMINED set SET \
... abort examined EXAMINED set SET \
... shutdown-complete examined EXAMINED set SET \
... outbound-interface [not] <string> \
... action STANDARD chain <leafref> reject REJECT \
... connmark \
... set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
... save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
... restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
... log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
... mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
... tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### description

A comment to describe the rule.

```
description <string>
```

### protocol

Match the protocol.

```
protocol [not] VALUE
```

### not

Invert the match.

```
not
```

**VALUE (mandatory)**

The protocol to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| icmp         | ICMP protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

**destination**

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

**address**

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | prefix: address and CIDR mask.   |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | prefix: address and CIDR mask.   |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**ipv4**

Match the fragment.

```
ipv4 [not] fragment
```

**not**

Invert the match.

```
not
```

**fragment (mandatory)**

Match if the packet is a fragment.

```
fragment
```

**icmp-type**

Match the packet ICMP type.

icmp-type [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The ICMP type to match.

VALUE

| VALUE values              | Description                |
|---------------------------|----------------------------|
| any                       | Any ICMP type.             |
| echo-request              | Echo request.              |
| echo-reply                | Echo reply.                |
| destination-unreachable   | Destination unreachable.   |
| network-unreachable       | Network unreachable.       |
| host-unreachable          | Host unreachable.          |
| protocol-unreachable      | Protocol unreachable.      |
| port-unreachable          | Port unreachable.          |
| fragmentation-needed      | Fragmentation needed.      |
| source-route-failed       | Source route failed.       |
| network-unknown           | Network unknown.           |
| host-unknown              | Host unknown.              |
| network-prohibited        | Network prohibited.        |
| host-prohibited           | Host prohibited.           |
| TOS-network-unreachable   | TOS network unreachable.   |
| TOS-host-unreachable      | TOS host unreachable.      |
| communication-prohibited  | Communication prohibited.  |
| host-precedence-violation | Host precedence violation. |
| precedence-cutoff         | Precedence cutoff.         |
| source-quench             | Source quench.             |
| redirect                  | Redirect.                  |
| network-redirect          | Network redirect.          |

continues on next page

Table 4 – continued from previous page

| VALUE values               | Description                        |
|----------------------------|------------------------------------|
| host-redirect              | Host redirect.                     |
| TOS-network-redirect       | TOS network redirect.              |
| TOS-host-redirect          | TOS host redirect.                 |
| router-advertisement       | Router advertisement.              |
| router-solicitation        | Router solicitation.               |
| ttl-exceeded               | TTL exceeded.                      |
| ttl-zero-during-transit    | Time to Live exceeded in Transit.  |
| ttl-zero-during-reassembly | Fragment Reassembly Time Exceeded. |
| parameter-problem          | Parameter problem.                 |
| ip-header-bad              | Bad IP header.                     |
| required-option-missing    | Missing a Required Option.         |
| timestamp-request          | Timestamp request.                 |
| timestamp-reply            | Timestamp reply.                   |
| address-mask-request       | Address mask request.              |
| address-mask-reply         | Address mask reply.                |

**tcp-flags**

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

**not**

Invert the match.

```
not
```

**set**

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

**contrack**

Match contrack information.

```
contrack \  
  status [not] VALUE \  
  state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \  
shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \  
forward-tsn \  
    data examined EXAMINED set SET \  
    abort examined EXAMINED set SET \  
    shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

abort examined EXAMINED set SET

### examined

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### outbound-interface

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

### action

The action performed by this rule.

```
action STANDARD chain <leafref> reject REJECT \
    connmark \
        set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
        save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
        restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
    mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <leafref>
```

### reject

Used to send back an error packet in response to the matched packet.

```
reject REJECT
```

| REJECT values          | Description   |
|------------------------|---|
| icmp-net-unreachable   | Reject with ICMP network unreachable.   |
| icmp-host-unreachable  | Reject with ICMP host unreachable.  |
| icmp-port-unreachable  | Reject with ICMP port unreachable.  |
| icmp-proto-unreachable | Reject with ICMP prototype unreachable.   |
| icmp-net-prohibited    | Reject with ICMP network prohibited.  |
| icmp-host-prohibited   | Reject with ICMP host prohibited.   |
| icmp-admin-prohibited  | Reject with ICMP admin prohibited.  |
| tcp-reset              | Reject with TCP RST packet. Can be used on rules which only match the TCP protocol. |

### connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### set-xmark

Zero out the bits given by mask and XOR value into the ctmask.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

### <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

### mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nfmak) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmak \& nfmask)$  i.e. ctmask defines what bits to clear and nfmak what bits of the nfmak to XOR into the ctmark. ctmask and nfmak default to 0xFFFFFFFF.

```
save-mark nfmak <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmak

Bits that should be XORed into the connection mark.

```
nfmak <0x0-0xffffffff>
```

## ctmask

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

## restore-mark

Copy the connection mark (ctmark) to the packet mark (nfmak) using the given masks. The new ctmark value is determined as follows:  $nfmak = (nfmak \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmak defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmak default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmak <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmak

Bits that should be cleared.

```
nfmak <0x0-0xffffffff>
```

### ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

### prefix

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

**tcpmss**

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

### clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

### counters (state only)

The counters of this rule.

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 filter output rule <uint64> counters_
↳packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 filter output rule <uint64> counters bytes
```

### chain

User chain.

```
vrouter running config# vrf <vrf> firewall ipv4 filter chain <string>
```

|          |                      |
|----------|----------------------|
| <string> | The user chain name. |
|----------|----------------------|

## policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv4 filter chain <string>
vrouter running chain <string># policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### Default value

accept

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 filter chain <string> packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 filter chain <string> bytes
```

## rule

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv4 filter chain <string>
vrouter running chain <string># rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
```

(continues on next page)

(continued from previous page)

```

... source \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... ipv4 [not] fragment \
... icmp-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
...   status [not] VALUE \
...   state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
...   rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... inbound-interface [not] <string> \
... outbound-interface [not] <string> \
... rpfilter invert true|false \
... action STANDARD chain <leafref> dscp DSCP reject REJECT \
...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu \
...   tos <0x0-0xff> mask <0x0-0xff>

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

**description**

A comment to describe the rule.

```
description <string>
```

**protocol**

Match the protocol.

```
protocol [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| icmp         | ICMP protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

## destination

Match on destination fields.

```
destination \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on destination address.

```
address [not] VALUE
```

## not

Invert the match.

```
not
```

## VALUE (mandatory)

The address to match.

```
VALUE
```

| VALUE       | Description  |
|-------------|--|
| val-ues     |  |
| <domain>    | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>   | IPv4 address.  |
| <A.B.C.D/M> | prefix: address and CIDR mask.   |

### port

Match on destination port.

port [not] VALUE

### not

Invert the match.

not

### VALUE (mandatory)

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

## port-range

Match on destination port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

## not

Invert the match.

```
not
```

## VALUE (mandatory)

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**source**

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

**address**

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The address to match.

```
VALUE
```

| VALUE       | Description  |
|-------------|--|
| val-ues     |  |
| <domain>    | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>   | IPv4 address.  |
| <A.B.C.D/M> | prefix: address and CIDR mask.   |

### port

Match on source port.

port [not] VALUE

### not

Invert the match.

not

### VALUE (mandatory)

The port to match.

VALUE

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

## port-range

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

### not

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**ipv4**

Match the fragment.

```
ipv4 [not] fragment
```

**not**

Invert the match.

```
not
```

**fragment (mandatory)**

Match if the packet is a fragment.

```
fragment
```

**icmp-type**

Match the packet ICMP type.

```
icmp-type [not] VALUE
```

**not**

Invert the match.

```
not
```

### VALUE (mandatory)

The ICMP type to match.

VALUE

| VALUE values               | Description                        |
|----------------------------|------------------------------------|
| any                        | Any ICMP type.                     |
| echo-request               | Echo request.                      |
| echo-reply                 | Echo reply.                        |
| destination-unreachable    | Destination unreachable.           |
| network-unreachable        | Network unreachable.               |
| host-unreachable           | Host unreachable.                  |
| protocol-unreachable       | Protocol unreachable.              |
| port-unreachable           | Port unreachable.                  |
| fragmentation-needed       | Fragmentation needed.              |
| source-route-failed        | Source route failed.               |
| network-unknown            | Network unknown.                   |
| host-unknown               | Host unknown.                      |
| network-prohibited         | Network prohibited.                |
| host-prohibited            | Host prohibited.                   |
| TOS-network-unreachable    | TOS network unreachable.           |
| TOS-host-unreachable       | TOS host unreachable.              |
| communication-prohibited   | Communication prohibited.          |
| host-precedence-violation  | Host precedence violation.         |
| precedence-cutoff          | Precedence cutoff.                 |
| source-quench              | Source quench.                     |
| redirect                   | Redirect.                          |
| network-redirect           | Network redirect.                  |
| host-redirect              | Host redirect.                     |
| TOS-network-redirect       | TOS network redirect.              |
| TOS-host-redirect          | TOS host redirect.                 |
| router-advertisement       | Router advertisement.              |
| router-solicitation        | Router solicitation.               |
| ttl-exceeded               | TTL exceeded.                      |
| ttl-zero-during-transit    | Time to Live exceeded in Transit.  |
| ttl-zero-during-reassembly | Fragment Reassembly Time Exceeded. |
| parameter-problem          | Parameter problem.                 |
| ip-header-bad              | Bad IP header.                     |
| required-option-missing    | Missing a Required Option.         |
| timestamp-request          | Timestamp request.                 |
| timestamp-reply            | Timestamp reply.                   |
| address-mask-request       | Address mask request.              |

continues on next page

Table 5 – continued from previous page

| VALUE values       | Description         |
|--------------------|---------------------|
| address-mask-reply | Address mask reply. |

**tcp-flags**

Match the packet TCP flags.

tcp-flags [not] set SET examined EXAMINED

**not**

Invert the match.

not

**set**

Set flags.

set SET

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

**examined**

Examined flags.

examined EXAMINED

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
  status [not] VALUE \  
  state [not] VALUE
```

### status

Match the connection status.

```
status [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE

The contrack status to match.

```
VALUE
```

| VALUE values | Description  |
|--------------|--|
| none         | No status.   |
| expected     | This is an expected connection (i.e. a contrack helper set it up). |
| seen_reply   | Contrack has seen packets in both directions.                      |
| assured      | Contrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.          |

**state**

Match the packet state regarding contrack.

state [not] VALUE

**not**

Invert the match.

not

**VALUE**

The packet states to match.

VALUE

| VALUE val-ues | Description  |
|---------------|--|
| invalid       | Packet is associated with no known connection.   |
| new           | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| estab-lished  | Packet is associated with a connection which has seen packets in both directions.  |
| related       | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked     | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat          | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat          | A virtual state, matching if the original destination differs from the reply source.   |

**connmark**

Matches the mark field associated with a connection.

```
connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

```
dscp [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The DSCP value to match.

|       |
|-------|
| VALUE |
|-------|

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

|                                      |
|--------------------------------------|
| tos [not] <0x0-0xff> mask <0x0-0xff> |
|--------------------------------------|

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**sctp-chunk-types**

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \
  shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \
  forward-tsn \
  data examined EXAMINED set SET \
  abort examined EXAMINED set SET \
  shutdown-complete examined EXAMINED set SET
```

**not**

Invert the match.

```
not
```

**SCOPE (mandatory)**

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

**init**

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

**abort**

ABORT chunk.

```
abort examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

**set**

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

```
shutdown-complete examined EXAMINED set SET
```

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

**<string> (mandatory)**

The interface to match.

```
<string>
```

**outbound-interface**

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

**not**

Invert the match.

```
not
```

**<string> (mandatory)**

The interface to match.

```
<string>
```

**rpfilter**

Performs a reverse path filter test on a packet. If a reply to the packet would be sent via the same interface that the packet arrived on, the packet will match.

```
rpfilter invert true|false
```

**invert**

This will invert the sense of the match. Instead of matching packets that passed the reverse path filter test, match those that have failed it.

```
invert true|false
```

**Default value**

false

### action

The action performed by this rule.

```
action STANDARD chain <leafref> dscp DSCP reject REJECT \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu \
  tos <0x0-0xff> mask <0x0-0xff>
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <leafref>
```

### dscp

Alters the value of the DSCP bits within the tos header of the IPv4 packet.

```
dscp DSCP
```

| DSCP values | Description   |
|-------------|---|
| <uint8>     | A differentiated services code point (DSCP) marking within the IP header. |
| af11        | AF11 (assured forwarding) class (10).                                     |
| af12        | AF12 (assured forwarding) class (12).                                     |
| af13        | AF13 (assured forwarding) class (14).                                     |
| af21        | AF21 (assured forwarding) class (18).                                     |
| af22        | AF22 (assured forwarding) class (20).                                     |
| af23        | AF23 (assured forwarding) class (22).                                     |
| af31        | AF31 (assured forwarding) class (26).                                     |
| af32        | AF32 (assured forwarding) class (28).                                     |
| af33        | AF33 (assured forwarding) class (30).                                     |
| af41        | AF41 (assured forwarding) class (34).                                     |
| af42        | AF42 (assured forwarding) class (36).                                     |
| af43        | AF43 (assured forwarding) class (38).                                     |
| be          | BE (best effort) class (0).   |
| cs0         | CS0 (class selector) class (0).   |
| cs1         | CS1 (class selector) class (8).   |
| cs2         | CS2 (class selector) class (16).  |
| cs3         | CS3 (class selector) class (24).  |
| cs4         | CS4 (class selector) class (32).  |
| cs5         | CS5 (class selector) class (40).  |
| cs6         | CS6 (class selector) class (48).  |
| cs7         | CS7 (class selector) class (56).  |
| ef          | EF (expedited forwarding) class (46).                                     |

## reject

Used to send back an error packet in response to the matched packet.

```
reject REJECT
```

| REJECT values          | Description   |
|------------------------|---|
| icmp-net-unreachable   | Reject with ICMP network unreachable.   |
| icmp-host-unreachable  | Reject with ICMP host unreachable.  |
| icmp-port-unreachable  | Reject with ICMP port unreachable.  |
| icmp-proto-unreachable | Reject with ICMP prototype unreachable.   |
| icmp-net-prohibited    | Reject with ICMP network prohibited.  |
| icmp-host-prohibited   | Reject with ICMP host prohibited.   |
| icmp-admin-prohibited  | Reject with ICMP admin prohibited.  |
| tcp-reset              | Reject with TCP RST packet. Can be used on rules which only match the TCP protocol. |

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

## <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nmask) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nmask \& nmask)$  i.e. ctmask defines what bits to clear and nmask what bits of the nmask to XOR into the ctmark. ctmask and nmask default to 0xFFFFFFFF.

```
save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## **nfmask**

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

## **ctmask**

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

## **restore-mark**

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $\text{nfmark} = (\text{nfmark} \& \sim\text{nfmask}) \wedge (\text{ctmark} \& \text{ctmask})$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## **nfmask**

Bits that should be cleared.

```
nfmask <0x0-0xffffffff>
```

## **ctmask**

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

## log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

## level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

## prefix

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

## additional-infos

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

## mark

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

### <0x0-0xffffffff> (mandatory)

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

### clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

**tos**

Alters the value of the tos header of the IPv4 packet.

```
tos <0x0-0xff> mask <0x0-0xff>
```

**<0x0-0xff> (mandatory)**

Bits that should be XORed into the tos.

```
<0x0-0xff>
```

**mask**

Zero the bits given by this mask in the tos.

```
mask <0x0-0xff>
```

**counters (state only)**

The counters of this rule.

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 filter chain <string> rule <uint64> ↵  
↪counters packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 filter chain <string> rule <uint64> ↵  
↪counters bytes
```

## ipv4 mangle

---

**Note:** requires a Turbo Router Network License.

---

Packet alteration table.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle
```

## prerouting

Altering packets as soon as they come in.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle prerouting
```

## policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle prerouting
vrouter running prerouting# policy POLICY
```

| POLICY val-<br>ues | Description   |
|--------------------|---|
| accept             | Let the packet through.   |
| drop               | Drop the packet.  |
| return             | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## Default value

accept

## packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle prerouting packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle prerouting bytes
```

**rule**

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle prerouting
vrouter running prerouting# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... source \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... ipv4 [not] fragment \
... icmp-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
...   status [not] VALUE \
...   state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
...   rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... inbound-interface [not] <string> \
... rpfILTER invert true|false \
... action STANDARD chain <leafref> dscp DSCP \
```

(continues on next page)

(continued from previous page)

```

...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu \
...   tos <0x0-0xff> mask <0x0-0xff>

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### description

A comment to describe the rule.

```
description <string>
```

### protocol

Match the protocol.

```
protocol [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| icmp         | ICMP protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

## destination

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

## address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | IPv4 prefix: address and CIDR mask.  |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE                            | Description  |
|----------------------------------|--|
| val-<br>ues<br><br><domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>                        | IPv4 address.  |
| <A.B.C.D/M>                      | prefix: address and CIDR mask.   |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**ipv4**

Match the fragment.

```
ipv4 [not] fragment
```

**not**

Invert the match.

```
not
```

**fragment (mandatory)**

Match if the packet is a fragment.

```
fragment
```

**icmp-type**

Match the packet ICMP type.

icmp-type [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The ICMP type to match.

VALUE

| VALUE values              | Description                |
|---------------------------|----------------------------|
| any                       | Any ICMP type.             |
| echo-request              | Echo request.              |
| echo-reply                | Echo reply.                |
| destination-unreachable   | Destination unreachable.   |
| network-unreachable       | Network unreachable.       |
| host-unreachable          | Host unreachable.          |
| protocol-unreachable      | Protocol unreachable.      |
| port-unreachable          | Port unreachable.          |
| fragmentation-needed      | Fragmentation needed.      |
| source-route-failed       | Source route failed.       |
| network-unknown           | Network unknown.           |
| host-unknown              | Host unknown.              |
| network-prohibited        | Network prohibited.        |
| host-prohibited           | Host prohibited.           |
| TOS-network-unreachable   | TOS network unreachable.   |
| TOS-host-unreachable      | TOS host unreachable.      |
| communication-prohibited  | Communication prohibited.  |
| host-precedence-violation | Host precedence violation. |
| precedence-cutoff         | Precedence cutoff.         |
| source-quench             | Source quench.             |
| redirect                  | Redirect.                  |
| network-redirect          | Network redirect.          |

continues on next page

Table 6 – continued from previous page

| VALUE values               | Description                        |
|----------------------------|------------------------------------|
| host-redirect              | Host redirect.                     |
| TOS-network-redirect       | TOS network redirect.              |
| TOS-host-redirect          | TOS host redirect.                 |
| router-advertisement       | Router advertisement.              |
| router-solicitation        | Router solicitation.               |
| ttl-exceeded               | TTL exceeded.                      |
| ttl-zero-during-transit    | Time to Live exceeded in Transit.  |
| ttl-zero-during-reassembly | Fragment Reassembly Time Exceeded. |
| parameter-problem          | Parameter problem.                 |
| ip-header-bad              | Bad IP header.                     |
| required-option-missing    | Missing a Required Option.         |
| timestamp-request          | Timestamp request.                 |
| timestamp-reply            | Timestamp reply.                   |
| address-mask-request       | Address mask request.              |
| address-mask-reply         | Address mask reply.                |

**tcp-flags**

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

**not**

Invert the match.

```
not
```

**set**

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
  status [not] VALUE \  
  state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \
  shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \
  forward-tsn \
  data examined EXAMINED set SET \
  abort examined EXAMINED set SET \
  shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

**abort**

ABORT chunk.

abort examined EXAMINED set SET

**examined**

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

**set**

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

**shutdown-complete**

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

## rpfilter

Performs a reverse path filter test on a packet. If a reply to the packet would be sent via the same interface that the packet arrived on, the packet will match.

```
rpfilter invert true|false
```

## invert

This will invert the sense of the match. Instead of matching packets that passed the reverse path filter test, match those that have failed it.

```
invert true|false
```

## Default value

false

## action

The action performed by this rule.

```
action STANDARD chain <leafref> dscp DSCP \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu \
  tos <0x0-0xff> mask <0x0-0xff>
```

## STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## chain

Jump to the user chain by this name.

```
chain <leafref>
```

## dscp

Alters the value of the DSCP bits within the tos header of the IPv4 packet.

```
dscp DSCP
```

| DSCP values | Description   |
|-------------|---|
| <uint8>     | A differentiated services code point (DSCP) marking within the IP header. |
| af11        | AF11 (assured forwarding) class (10).                                     |
| af12        | AF12 (assured forwarding) class (12).                                     |
| af13        | AF13 (assured forwarding) class (14).                                     |
| af21        | AF21 (assured forwarding) class (18).                                     |
| af22        | AF22 (assured forwarding) class (20).                                     |
| af23        | AF23 (assured forwarding) class (22).                                     |
| af31        | AF31 (assured forwarding) class (26).                                     |
| af32        | AF32 (assured forwarding) class (28).                                     |
| af33        | AF33 (assured forwarding) class (30).                                     |
| af41        | AF41 (assured forwarding) class (34).                                     |
| af42        | AF42 (assured forwarding) class (36).                                     |
| af43        | AF43 (assured forwarding) class (38).                                     |
| be          | BE (best effort) class (0).   |
| cs0         | CS0 (class selector) class (0).   |
| cs1         | CS1 (class selector) class (8).   |
| cs2         | CS2 (class selector) class (16).  |
| cs3         | CS3 (class selector) class (24).  |
| cs4         | CS4 (class selector) class (32).  |
| cs5         | CS5 (class selector) class (40).  |
| cs6         | CS6 (class selector) class (48).  |
| cs7         | CS7 (class selector) class (56).  |
| ef          | EF (expedited forwarding) class (46).                                     |

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \  
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \  
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \  
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

XOR with this value.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

**save-mark**

Copy the packet mark (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmark \& nfmask)$  i.e. ctmask defines what bits to clear and nfmask what bits of the nfmark to XOR into the ctmark. ctmask and nfmask default to 0xFFFFFFFF.

```
save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

**nfmask**

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

**ctmask**

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

**restore-mark**

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $nfmark = (nfmark \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### nmask

Bits that should be cleared.

```
nmask <0x0-0xffffffff>
```

### ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

**prefix**

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

## set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

## clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

## tos

Alters the value of the tos header of the IPv4 packet.

```
tos <0x0-0xff> mask <0x0-0xff>
```

### <0x0-0xff> (mandatory)

Bits that should be XORed into the tos.

```
<0x0-0xff>
```

## mask

Zero the bits given by this mask in the tos.

```
mask <0x0-0xff>
```

### counters (state only)

The counters of this rule.

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle prerouting rule <uint64> counters
↳packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle prerouting rule <uint64> counters
↳bytes
```

### input

Altering packets before routing.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle input
```

### policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle input
vrouter running input# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### Default value

accept

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle input packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle input bytes
```

**rule**

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle input
vrouter running input# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... source \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... ipv4 [not] fragment \
... icmp-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... contrack \
... status [not] VALUE \
... state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
... rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
```

(continues on next page)

(continued from previous page)

```

... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
... data examined EXAMINED set SET \
... abort examined EXAMINED set SET \
... shutdown-complete examined EXAMINED set SET \
... inbound-interface [not] <string> \
... action STANDARD chain <leafref> dscp DSCP \
... connmark \
... set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
... save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
... restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
... log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
... mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
... tcpmss set-mss <uint32> clamp-mss-to-pmtu \
... tos <0x0-0xff> mask <0x0-0xff>

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

## description

A comment to describe the rule.

```
description <string>
```

## protocol

Match the protocol.

```
protocol [not] VALUE
```

## not

Invert the match.

```
not
```

### VALUE (mandatory)

The protocol to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| icmp         | ICMP protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |

### destination

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

### address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | prefix: address and CIDR mask.   |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE                            | Description  |
|----------------------------------|--|
| val-<br>ues<br><br><domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>                        | IPv4 address.  |
| <A.B.C.D/M>                      | prefix: address and CIDR mask.   |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**ipv4**

Match the fragment.

```
ipv4 [not] fragment
```

**not**

Invert the match.

```
not
```

**fragment (mandatory)**

Match if the packet is a fragment.

```
fragment
```

**icmp-type**

Match the packet ICMP type.

icmp-type [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The ICMP type to match.

VALUE

| VALUE values              | Description                |
|---------------------------|----------------------------|
| any                       | Any ICMP type.             |
| echo-request              | Echo request.              |
| echo-reply                | Echo reply.                |
| destination-unreachable   | Destination unreachable.   |
| network-unreachable       | Network unreachable.       |
| host-unreachable          | Host unreachable.          |
| protocol-unreachable      | Protocol unreachable.      |
| port-unreachable          | Port unreachable.          |
| fragmentation-needed      | Fragmentation needed.      |
| source-route-failed       | Source route failed.       |
| network-unknown           | Network unknown.           |
| host-unknown              | Host unknown.              |
| network-prohibited        | Network prohibited.        |
| host-prohibited           | Host prohibited.           |
| TOS-network-unreachable   | TOS network unreachable.   |
| TOS-host-unreachable      | TOS host unreachable.      |
| communication-prohibited  | Communication prohibited.  |
| host-precedence-violation | Host precedence violation. |
| precedence-cutoff         | Precedence cutoff.         |
| source-quench             | Source quench.             |
| redirect                  | Redirect.                  |
| network-redirect          | Network redirect.          |

continues on next page

Table 7 – continued from previous page

| VALUE values               | Description                        |
|----------------------------|------------------------------------|
| host-redirect              | Host redirect.                     |
| TOS-network-redirect       | TOS network redirect.              |
| TOS-host-redirect          | TOS host redirect.                 |
| router-advertisement       | Router advertisement.              |
| router-solicitation        | Router solicitation.               |
| ttl-exceeded               | TTL exceeded.                      |
| ttl-zero-during-transit    | Time to Live exceeded in Transit.  |
| ttl-zero-during-reassembly | Fragment Reassembly Time Exceeded. |
| parameter-problem          | Parameter problem.                 |
| ip-header-bad              | Bad IP header.                     |
| required-option-missing    | Missing a Required Option.         |
| timestamp-request          | Timestamp request.                 |
| timestamp-reply            | Timestamp reply.                   |
| address-mask-request       | Address mask request.              |
| address-mask-reply         | Address mask reply.                |

**tcp-flags**

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

**not**

Invert the match.

```
not
```

**set**

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
  status [not] VALUE \  
  state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \  
shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \  
forward-tsn \  
    data examined EXAMINED set SET \  
    abort examined EXAMINED set SET \  
    shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

abort examined EXAMINED set SET

### examined

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

### action

The action performed by this rule.

```
action STANDARD chain <leafref> dscp DSCP \
    connmark \
        set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
        save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
        restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
    mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    tcpmss set-mss <uint32> clamp-mss-to-pmtu \
    tos <0x0-0xff> mask <0x0-0xff>
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <leafref>
```

### dscp

Alters the value of the DSCP bits within the tos header of the IPv4 packet.

```
dscp DSCP
```

| DSCP values | Description   |
|-------------|---|
| <uint8>     | A differentiated services code point (DSCP) marking within the IP header. |
| af11        | AF11 (assured forwarding) class (10).                                     |
| af12        | AF12 (assured forwarding) class (12).                                     |
| af13        | AF13 (assured forwarding) class (14).                                     |
| af21        | AF21 (assured forwarding) class (18).                                     |
| af22        | AF22 (assured forwarding) class (20).                                     |
| af23        | AF23 (assured forwarding) class (22).                                     |
| af31        | AF31 (assured forwarding) class (26).                                     |
| af32        | AF32 (assured forwarding) class (28).                                     |
| af33        | AF33 (assured forwarding) class (30).                                     |
| af41        | AF41 (assured forwarding) class (34).                                     |
| af42        | AF42 (assured forwarding) class (36).                                     |
| af43        | AF43 (assured forwarding) class (38).                                     |
| be          | BE (best effort) class (0).   |
| cs0         | CS0 (class selector) class (0).   |
| cs1         | CS1 (class selector) class (8).   |
| cs2         | CS2 (class selector) class (16).  |
| cs3         | CS3 (class selector) class (24).  |
| cs4         | CS4 (class selector) class (32).  |
| cs5         | CS5 (class selector) class (40).  |
| cs6         | CS6 (class selector) class (48).  |
| cs7         | CS7 (class selector) class (56).  |
| ef          | EF (expedited forwarding) class (46).                                     |

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

XOR with this value.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

**save-mark**

Copy the packet mark (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmark \& nfmask)$  i.e. ctmask defines what bits to clear and nfmask what bits of the nfmark to XOR into the ctmark. ctmask and nfmask default to 0xFFFFFFFF.

```
save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

**nfmask**

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

**ctmask**

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

**restore-mark**

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $nfmark = (nfmark \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### nmask

Bits that should be cleared.

```
nmask <0x0-0xffffffff>
```

### ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

**prefix**

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

## set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

## clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

## tos

Alters the value of the tos header of the IPv4 packet.

```
tos <0x0-0xff> mask <0x0-0xff>
```

### <0x0-0xff> (mandatory)

Bits that should be XORed into the tos.

```
<0x0-0xff>
```

## mask

Zero the bits given by this mask in the tos.

```
mask <0x0-0xff>
```

**counters (state only)**

The counters of this rule.

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle input rule <uint64> counters packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle input rule <uint64> counters bytes
```

**forward**

Altering packets being routed.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle forward
```

**policy**

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle forward
vrouter running forward# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

**Default value**

accept

## packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle forward packets
```

## bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle forward bytes
```

## rule

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle forward
vrouter running forward# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... source \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... ipv4 [not] fragment \
... icmp-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
... status [not] VALUE \
... state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
... rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
```

(continues on next page)

(continued from previous page)

```

... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
... data examined EXAMINED set SET \
... abort examined EXAMINED set SET \
... shutdown-complete examined EXAMINED set SET \
... inbound-interface [not] <string> \
... outbound-interface [not] <string> \
... action STANDARD chain <leafref> dscp DSCP \
... connmark \
... set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
... save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
... restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
... log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
... mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
... tcpmss set-mss <uint32> clamp-mss-to-pmtu \
... tos <0x0-0xff> mask <0x0-0xff>

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

## description

A comment to describe the rule.

```
description <string>
```

## protocol

Match the protocol.

```
protocol [not] VALUE
```

## not

Invert the match.

```
not
```

### VALUE (mandatory)

The protocol to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| icmp         | ICMP protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |

### destination

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

### address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE                            | Description  |
|----------------------------------|--|
| val-<br>ues<br><br><domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>                        | IPv4 address.  |
| <A.B.C.D/M>                      | prefix: address and CIDR mask.   |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | prefix: address and CIDR mask.   |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**ipv4**

Match the fragment.

```
ipv4 [not] fragment
```

**not**

Invert the match.

```
not
```

**fragment (mandatory)**

Match if the packet is a fragment.

```
fragment
```

**icmp-type**

Match the packet ICMP type.

icmp-type [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The ICMP type to match.

VALUE

| VALUE values              | Description                |
|---------------------------|----------------------------|
| any                       | Any ICMP type.             |
| echo-request              | Echo request.              |
| echo-reply                | Echo reply.                |
| destination-unreachable   | Destination unreachable.   |
| network-unreachable       | Network unreachable.       |
| host-unreachable          | Host unreachable.          |
| protocol-unreachable      | Protocol unreachable.      |
| port-unreachable          | Port unreachable.          |
| fragmentation-needed      | Fragmentation needed.      |
| source-route-failed       | Source route failed.       |
| network-unknown           | Network unknown.           |
| host-unknown              | Host unknown.              |
| network-prohibited        | Network prohibited.        |
| host-prohibited           | Host prohibited.           |
| TOS-network-unreachable   | TOS network unreachable.   |
| TOS-host-unreachable      | TOS host unreachable.      |
| communication-prohibited  | Communication prohibited.  |
| host-precedence-violation | Host precedence violation. |
| precedence-cutoff         | Precedence cutoff.         |
| source-quench             | Source quench.             |
| redirect                  | Redirect.                  |
| network-redirect          | Network redirect.          |

continues on next page

Table 8 – continued from previous page

| VALUE values               | Description                        |
|----------------------------|------------------------------------|
| host-redirect              | Host redirect.                     |
| TOS-network-redirect       | TOS network redirect.              |
| TOS-host-redirect          | TOS host redirect.                 |
| router-advertisement       | Router advertisement.              |
| router-solicitation        | Router solicitation.               |
| ttl-exceeded               | TTL exceeded.                      |
| ttl-zero-during-transit    | Time to Live exceeded in Transit.  |
| ttl-zero-during-reassembly | Fragment Reassembly Time Exceeded. |
| parameter-problem          | Parameter problem.                 |
| ip-header-bad              | Bad IP header.                     |
| required-option-missing    | Missing a Required Option.         |
| timestamp-request          | Timestamp request.                 |
| timestamp-reply            | Timestamp reply.                   |
| address-mask-request       | Address mask request.              |
| address-mask-reply         | Address mask reply.                |

**tcp-flags**

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

**not**

Invert the match.

```
not
```

**set**

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
  status [not] VALUE \  
  state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

## sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \
  shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \
  forward-tsn \
  data examined EXAMINED set SET \
  abort examined EXAMINED set SET \
  shutdown-complete examined EXAMINED set SET
```

### not

Invert the match.

```
not
```

### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

**abort**

ABORT chunk.

abort examined EXAMINED set SET

**examined**

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

**set**

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

**shutdown-complete**

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

**outbound-interface**

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

**not**

Invert the match.

```
not
```

**<string> (mandatory)**

The interface to match.

```
<string>
```

**action**

The action performed by this rule.

```
action STANDARD chain <leafref> dscp DSCP \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu \
  tos <0x0-0xff> mask <0x0-0xff>
```

**STANDARD**

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## chain

Jump to the user chain by this name.

```
chain <leafref>
```

## dscp

Alters the value of the DSCP bits within the tos header of the IPv4 packet.

```
dscp DSCP
```

| DSCP values | Description   |
|-------------|---|
| <uint8>     | A differentiated services code point (DSCP) marking within the IP header. |
| af11        | AF11 (assured forwarding) class (10).                                     |
| af12        | AF12 (assured forwarding) class (12).                                     |
| af13        | AF13 (assured forwarding) class (14).                                     |
| af21        | AF21 (assured forwarding) class (18).                                     |
| af22        | AF22 (assured forwarding) class (20).                                     |
| af23        | AF23 (assured forwarding) class (22).                                     |
| af31        | AF31 (assured forwarding) class (26).                                     |
| af32        | AF32 (assured forwarding) class (28).                                     |
| af33        | AF33 (assured forwarding) class (30).                                     |
| af41        | AF41 (assured forwarding) class (34).                                     |
| af42        | AF42 (assured forwarding) class (36).                                     |
| af43        | AF43 (assured forwarding) class (38).                                     |
| be          | BE (best effort) class (0).   |
| cs0         | CS0 (class selector) class (0).   |
| cs1         | CS1 (class selector) class (8).   |
| cs2         | CS2 (class selector) class (16).  |
| cs3         | CS3 (class selector) class (24).  |
| cs4         | CS4 (class selector) class (32).  |
| cs5         | CS5 (class selector) class (40).  |
| cs6         | CS6 (class selector) class (48).  |
| cs7         | CS7 (class selector) class (56).  |
| ef          | EF (expedited forwarding) class (46).                                     |

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

XOR with this value.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

**save-mark**

Copy the packet mark (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmark \& nfmask)$  i.e. ctmask defines what bits to clear and nfmask what bits of the nfmark to XOR into the ctmark. ctmask and nfmask default to 0xFFFFFFFF.

```
save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

**nfmask**

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

**ctmask**

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

**restore-mark**

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $nfmark = (nfmark \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### nmask

Bits that should be cleared.

```
nmask <0x0-0xffffffff>
```

### ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

**prefix**

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

## set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

## clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

## tos

Alters the value of the tos header of the IPv4 packet.

```
tos <0x0-0xff> mask <0x0-0xff>
```

### <0x0-0xff> (mandatory)

Bits that should be XORed into the tos.

```
<0x0-0xff>
```

## mask

Zero the bits given by this mask in the tos.

```
mask <0x0-0xff>
```

**counters (state only)**

The counters of this rule.

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle forward rule <uint64> counters
↳packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle forward rule <uint64> counters bytes
```

**output**

Altering locally-generated packets before routing.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle output
```

**policy**

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle output
vrouter running output# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

**Default value**

accept

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle output packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle output bytes
```

**rule**

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle output
vrouter running output# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... source \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... ipv4 [not] fragment \
... icmp-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... contrack \
... status [not] VALUE \
... state [not] VALUE \
... conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
... rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
```

(continues on next page)

(continued from previous page)

```

... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
... data examined EXAMINED set SET \
... abort examined EXAMINED set SET \
... shutdown-complete examined EXAMINED set SET \
... outbound-interface [not] <string> \
... action STANDARD chain <leafref> dscp DSCP \
... connmark \
... set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
... save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
... restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
... log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
... mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
... tcpmss set-mss <uint32> clamp-mss-to-pmtu \
... tos <0x0-0xff> mask <0x0-0xff>

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### description

A comment to describe the rule.

```
description <string>
```

### protocol

Match the protocol.

```
protocol [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

The protocol to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| icmp         | ICMP protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |

### destination

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

### address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE                            | Description  |
|----------------------------------|--|
| val-<br>ues<br><br><domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>                        | IPv4 address.  |
| <A.B.C.D/M>                      | IPv4 prefix: address and CIDR mask.  |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE                            | Description  |
|----------------------------------|--|
| val-<br>ues<br><br><domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>                        | IPv4 address.  |
| <A.B.C.D/M>                      | prefix: address and CIDR mask.   |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**ipv4**

Match the fragment.

```
ipv4 [not] fragment
```

**not**

Invert the match.

```
not
```

**fragment (mandatory)**

Match if the packet is a fragment.

```
fragment
```

### icmp-type

Match the packet ICMP type.

icmp-type [not] VALUE

### not

Invert the match.

not

### VALUE (mandatory)

The ICMP type to match.

VALUE

| VALUE values              | Description                |
|---------------------------|----------------------------|
| any                       | Any ICMP type.             |
| echo-request              | Echo request.              |
| echo-reply                | Echo reply.                |
| destination-unreachable   | Destination unreachable.   |
| network-unreachable       | Network unreachable.       |
| host-unreachable          | Host unreachable.          |
| protocol-unreachable      | Protocol unreachable.      |
| port-unreachable          | Port unreachable.          |
| fragmentation-needed      | Fragmentation needed.      |
| source-route-failed       | Source route failed.       |
| network-unknown           | Network unknown.           |
| host-unknown              | Host unknown.              |
| network-prohibited        | Network prohibited.        |
| host-prohibited           | Host prohibited.           |
| TOS-network-unreachable   | TOS network unreachable.   |
| TOS-host-unreachable      | TOS host unreachable.      |
| communication-prohibited  | Communication prohibited.  |
| host-precedence-violation | Host precedence violation. |
| precedence-cutoff         | Precedence cutoff.         |
| source-quench             | Source quench.             |
| redirect                  | Redirect.                  |
| network-redirect          | Network redirect.          |

continues on next page

Table 9 – continued from previous page

| VALUE values               | Description                        |
|----------------------------|------------------------------------|
| host-redirect              | Host redirect.                     |
| TOS-network-redirect       | TOS network redirect.              |
| TOS-host-redirect          | TOS host redirect.                 |
| router-advertisement       | Router advertisement.              |
| router-solicitation        | Router solicitation.               |
| ttl-exceeded               | TTL exceeded.                      |
| ttl-zero-during-transit    | Time to Live exceeded in Transit.  |
| ttl-zero-during-reassembly | Fragment Reassembly Time Exceeded. |
| parameter-problem          | Parameter problem.                 |
| ip-header-bad              | Bad IP header.                     |
| required-option-missing    | Missing a Required Option.         |
| timestamp-request          | Timestamp request.                 |
| timestamp-reply            | Timestamp reply.                   |
| address-mask-request       | Address mask request.              |
| address-mask-reply         | Address mask reply.                |

**tcp-flags**

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

**not**

Invert the match.

```
not
```

**set**

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

**contrack**

Match contrack information.

```
contrack \  
  status [not] VALUE \  
  state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \
  shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \
  forward-tsn \
  data examined EXAMINED set SET \
  abort examined EXAMINED set SET \
  shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

abort examined EXAMINED set SET

### examined

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### outbound-interface

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

### action

The action performed by this rule.

```
action STANDARD chain <leafref> dscp DSCP \
    connmark \
        set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
        save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
        restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
    mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    tcpmss set-mss <uint32> clamp-mss-to-pmtu \
    tos <0x0-0xff> mask <0x0-0xff>
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <leafref>
```

### dscp

Alters the value of the DSCP bits within the tos header of the IPv4 packet.

```
dscp DSCP
```

| DSCP values | Description   |
|-------------|---|
| <uint8>     | A differentiated services code point (DSCP) marking within the IP header. |
| af11        | AF11 (assured forwarding) class (10).                                     |
| af12        | AF12 (assured forwarding) class (12).                                     |
| af13        | AF13 (assured forwarding) class (14).                                     |
| af21        | AF21 (assured forwarding) class (18).                                     |
| af22        | AF22 (assured forwarding) class (20).                                     |
| af23        | AF23 (assured forwarding) class (22).                                     |
| af31        | AF31 (assured forwarding) class (26).                                     |
| af32        | AF32 (assured forwarding) class (28).                                     |
| af33        | AF33 (assured forwarding) class (30).                                     |
| af41        | AF41 (assured forwarding) class (34).                                     |
| af42        | AF42 (assured forwarding) class (36).                                     |
| af43        | AF43 (assured forwarding) class (38).                                     |
| be          | BE (best effort) class (0).   |
| cs0         | CS0 (class selector) class (0).   |
| cs1         | CS1 (class selector) class (8).   |
| cs2         | CS2 (class selector) class (16).  |
| cs3         | CS3 (class selector) class (24).  |
| cs4         | CS4 (class selector) class (32).  |
| cs5         | CS5 (class selector) class (40).  |
| cs6         | CS6 (class selector) class (48).  |
| cs7         | CS7 (class selector) class (56).  |
| ef          | EF (expedited forwarding) class (46).                                     |

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

XOR with this value.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

**save-mark**

Copy the packet mark (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmark \& nfmask)$  i.e. ctmask defines what bits to clear and nfmask what bits of the nfmark to XOR into the ctmark. ctmask and nfmask default to 0xFFFFFFFF.

```
save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

**nfmask**

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

**ctmask**

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

**restore-mark**

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $nfmark = (nfmark \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### nmask

Bits that should be cleared.

```
nmask <0x0-0xffffffff>
```

### ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

**prefix**

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

## set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

## clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

## tos

Alters the value of the tos header of the IPv4 packet.

```
tos <0x0-0xff> mask <0x0-0xff>
```

### <0x0-0xff> (mandatory)

Bits that should be XORed into the tos.

```
<0x0-0xff>
```

## mask

Zero the bits given by this mask in the tos.

```
mask <0x0-0xff>
```

**counters (state only)**

The counters of this rule.

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle output rule <uint64> counters_
↳packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle output rule <uint64> counters bytes
```

**postrouting**

Altering packets as they are about to go.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle postrouting
```

**policy**

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle postrouting
vrouter running postrouting# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

**Default value**

accept

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle postrouting packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle postrouting bytes
```

**rule**

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle postrouting
vrouter running postrouting# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... source \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... ipv4 [not] fragment \
... icmp-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
... status [not] VALUE \
... state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
... rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
```

(continues on next page)

(continued from previous page)

```

... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
... data examined EXAMINED set SET \
... abort examined EXAMINED set SET \
... shutdown-complete examined EXAMINED set SET \
... outbound-interface [not] <string> \
... action STANDARD chain <leafref> dscp DSCP \
... connmark \
... set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
... save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
... restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
... log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
... mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
... tcpmss set-mss <uint32> clamp-mss-to-pmtu \
... tos <0x0-0xff> mask <0x0-0xff>

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### description

A comment to describe the rule.

```
description <string>
```

### protocol

Match the protocol.

```
protocol [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

The protocol to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| icmp         | ICMP protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |

### destination

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

### address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | prefix: address and CIDR mask.   |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C/DIM>   | prefix: address and CIDR mask.   |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**ipv4**

Match the fragment.

```
ipv4 [not] fragment
```

**not**

Invert the match.

```
not
```

**fragment (mandatory)**

Match if the packet is a fragment.

```
fragment
```

### icmp-type

Match the packet ICMP type.

icmp-type [not] VALUE

### not

Invert the match.

not

### VALUE (mandatory)

The ICMP type to match.

VALUE

| VALUE values              | Description                |
|---------------------------|----------------------------|
| any                       | Any ICMP type.             |
| echo-request              | Echo request.              |
| echo-reply                | Echo reply.                |
| destination-unreachable   | Destination unreachable.   |
| network-unreachable       | Network unreachable.       |
| host-unreachable          | Host unreachable.          |
| protocol-unreachable      | Protocol unreachable.      |
| port-unreachable          | Port unreachable.          |
| fragmentation-needed      | Fragmentation needed.      |
| source-route-failed       | Source route failed.       |
| network-unknown           | Network unknown.           |
| host-unknown              | Host unknown.              |
| network-prohibited        | Network prohibited.        |
| host-prohibited           | Host prohibited.           |
| TOS-network-unreachable   | TOS network unreachable.   |
| TOS-host-unreachable      | TOS host unreachable.      |
| communication-prohibited  | Communication prohibited.  |
| host-precedence-violation | Host precedence violation. |
| precedence-cutoff         | Precedence cutoff.         |
| source-quench             | Source quench.             |
| redirect                  | Redirect.                  |
| network-redirect          | Network redirect.          |

continues on next page

Table 10 – continued from previous page

| VALUE values               | Description                        |
|----------------------------|------------------------------------|
| host-redirect              | Host redirect.                     |
| TOS-network-redirect       | TOS network redirect.              |
| TOS-host-redirect          | TOS host redirect.                 |
| router-advertisement       | Router advertisement.              |
| router-solicitation        | Router solicitation.               |
| ttl-exceeded               | TTL exceeded.                      |
| ttl-zero-during-transit    | Time to Live exceeded in Transit.  |
| ttl-zero-during-reassembly | Fragment Reassembly Time Exceeded. |
| parameter-problem          | Parameter problem.                 |
| ip-header-bad              | Bad IP header.                     |
| required-option-missing    | Missing a Required Option.         |
| timestamp-request          | Timestamp request.                 |
| timestamp-reply            | Timestamp reply.                   |
| address-mask-request       | Address mask request.              |
| address-mask-reply         | Address mask reply.                |

**tcp-flags**

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

**not**

Invert the match.

```
not
```

**set**

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

**contrack**

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \  
shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \  
forward-tsn \  
    data examined EXAMINED set SET \  
    abort examined EXAMINED set SET \  
    shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

asconf-ack

**forward-tsn**

FORWARD TSN chunk.

forward-tsn

**data**

DATA chunk.

data examined EXAMINED set SET

**examined**

Examined flags.

examined EXAMINED

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

set SET

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

**abort**

ABORT chunk.

abort examined EXAMINED set SET

**examined**

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

**set**

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

**shutdown-complete**

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### outbound-interface

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

### action

The action performed by this rule.

```
action STANDARD chain <leafref> dscp DSCP \
    connmark \
        set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
        save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
        restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
    mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    tcpmss set-mss <uint32> clamp-mss-to-pmtu \
    tos <0x0-0xff> mask <0x0-0xff>
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <leafref>
```

### dscp

Alters the value of the DSCP bits within the tos header of the IPv4 packet.

```
dscp DSCP
```

| DSCP values | Description   |
|-------------|---|
| <uint8>     | A differentiated services code point (DSCP) marking within the IP header. |
| af11        | AF11 (assured forwarding) class (10).                                     |
| af12        | AF12 (assured forwarding) class (12).                                     |
| af13        | AF13 (assured forwarding) class (14).                                     |
| af21        | AF21 (assured forwarding) class (18).                                     |
| af22        | AF22 (assured forwarding) class (20).                                     |
| af23        | AF23 (assured forwarding) class (22).                                     |
| af31        | AF31 (assured forwarding) class (26).                                     |
| af32        | AF32 (assured forwarding) class (28).                                     |
| af33        | AF33 (assured forwarding) class (30).                                     |
| af41        | AF41 (assured forwarding) class (34).                                     |
| af42        | AF42 (assured forwarding) class (36).                                     |
| af43        | AF43 (assured forwarding) class (38).                                     |
| be          | BE (best effort) class (0).   |
| cs0         | CS0 (class selector) class (0).   |
| cs1         | CS1 (class selector) class (8).   |
| cs2         | CS2 (class selector) class (16).  |
| cs3         | CS3 (class selector) class (24).  |
| cs4         | CS4 (class selector) class (32).  |
| cs5         | CS5 (class selector) class (40).  |
| cs6         | CS6 (class selector) class (48).  |
| cs7         | CS7 (class selector) class (56).  |
| ef          | EF (expedited forwarding) class (46).                                     |

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

XOR with this value.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

**save-mark**

Copy the packet mark (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmark \& nfmask)$  i.e. ctmask defines what bits to clear and nfmask what bits of the nfmark to XOR into the ctmark. ctmask and nfmask default to 0xFFFFFFFF.

```
save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

**nfmask**

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

**ctmask**

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

**restore-mark**

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $nfmark = (nfmark \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### nmask

Bits that should be cleared.

```
nmask <0x0-0xffffffff>
```

### ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

**prefix**

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

## set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

## clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

## tos

Alters the value of the tos header of the IPv4 packet.

```
tos <0x0-0xff> mask <0x0-0xff>
```

### <0x0-0xff> (mandatory)

Bits that should be XORed into the tos.

```
<0x0-0xff>
```

## mask

Zero the bits given by this mask in the tos.

```
mask <0x0-0xff>
```

### counters (state only)

The counters of this rule.

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle postrouting rule <uint64> counters_
↳packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle postrouting rule <uint64> counters_
↳bytes
```

### chain

User chain.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle chain <string>
```

|          |                      |
|----------|----------------------|
| <string> | The user chain name. |
|----------|----------------------|

### policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle chain <string>
vrouter running chain <string># policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

**Default value**

accept

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle chain <string> packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle chain <string> bytes
```

**rule**

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv4 mangle chain <string>
vrouter running chain <string># rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... source \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... ipv4 [not] fragment \
... icmp-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
...   status [not] VALUE \
...   state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
...   rate <uint32> UNIT \
```

(continues on next page)

(continued from previous page)

```

... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... inbound-interface [not] <string> \
... outbound-interface [not] <string> \
... rpfILTER invert true|false \
... action STANDARD chain <leafref> dscp DSCP reject REJECT \
...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-Infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu \
...   tos <0x0-0xff> mask <0x0-0xff>

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### description

A comment to describe the rule.

```
description <string>
```

### protocol

Match the protocol.

```
protocol [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| icmp         | ICMP protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

**destination**

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

**address**

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The address to match.

```
VALUE
```

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | IPv4 prefix: address and CIDR mask.  |

**port**

Match on destination port.

```
port [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on destination port range (syntax: port[,port|port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

### VALUE (mandatory)

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

### group

Matches a set of addresses or networks.

```
group [not] <string>
```

### not

Not match-set.

```
not
```

### <string> (mandatory)

The name of the group.

```
<string>
```

### source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

### address

Match on source address.

```
address [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

The address to match.

```
VALUE
```

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | IPv4 prefix: address and CIDR mask.  |

**port**

Match on source port.

```
port [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

```
VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.
```

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**ipv4**

Match the fragment.

```
ipv4 [not] fragment
```

**not**

Invert the match.

```
not
```

**fragment (mandatory)**

Match if the packet is a fragment.

```
fragment
```

**icmp-type**

Match the packet ICMP type.

```
icmp-type [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The ICMP type to match.

```
VALUE
```

| VALUE values            | Description              |
|-------------------------|--------------------------|
| any                     | Any ICMP type.           |
| echo-request            | Echo request.            |
| echo-reply              | Echo reply.              |
| destination-unreachable | Destination unreachable. |
| network-unreachable     | Network unreachable.     |
| host-unreachable        | Host unreachable.        |
| protocol-unreachable    | Protocol unreachable.    |
| port-unreachable        | Port unreachable.        |

continues on next page

Table 11 – continued from previous page

| VALUE values               | Description                        |
|----------------------------|------------------------------------|
| fragmentation-needed       | Fragmentation needed.              |
| source-route-failed        | Source route failed.               |
| network-unknown            | Network unknown.                   |
| host-unknown               | Host unknown.                      |
| network-prohibited         | Network prohibited.                |
| host-prohibited            | Host prohibited.                   |
| TOS-network-unreachable    | TOS network unreachable.           |
| TOS-host-unreachable       | TOS host unreachable.              |
| communication-prohibited   | Communication prohibited.          |
| host-precedence-violation  | Host precedence violation.         |
| precedence-cutoff          | Precedence cutoff.                 |
| source-quench              | Source quench.                     |
| redirect                   | Redirect.                          |
| network-redirect           | Network redirect.                  |
| host-redirect              | Host redirect.                     |
| TOS-network-redirect       | TOS network redirect.              |
| TOS-host-redirect          | TOS host redirect.                 |
| router-advertisement       | Router advertisement.              |
| router-solicitation        | Router solicitation.               |
| ttl-exceeded               | TTL exceeded.                      |
| ttl-zero-during-transit    | Time to Live exceeded in Transit.  |
| ttl-zero-during-reassembly | Fragment Reassembly Time Exceeded. |
| parameter-problem          | Parameter problem.                 |
| ip-header-bad              | Bad IP header.                     |
| required-option-missing    | Missing a Required Option.         |
| timestamp-request          | Timestamp request.                 |
| timestamp-reply            | Timestamp reply.                   |
| address-mask-request       | Address mask request.              |
| address-mask-reply         | Address mask reply.                |

### tcp-flags

Match the packet TCP flags.

|   |
|---|
| tcp-flags [not] set SET examined EXAMINED |
|---|

**not**

Invert the match.

```
not
```

**set**

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

### status

Match the connection status.

```
status [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE

The contrack status to match.

```
VALUE
```

| VALUE values | Description  |
|--------------|--|
| none         | No status.   |
| expected     | This is an expected connection (i.e. a contrack helper set it up). |
| seen_reply   | Contrack has seen packets in both directions.                      |
| assured      | Contrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.          |

**state**

Match the packet state regarding conntrack.

```
state [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE**

The packet states to match.

```
VALUE
```

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

**connmark**

Matches the mark field associated with a connection.

```
connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

<uint32>

**UNIT**

Unit for rate.

UNIT

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \
  shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \
  forward-tsn \
  data examined EXAMINED set SET \
  abort examined EXAMINED set SET \
  shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

asconf-ack

**forward-tsn**

FORWARD TSN chunk.

forward-tsn

**data**

DATA chunk.

data examined EXAMINED set SET

**examined**

Examined flags.

examined EXAMINED

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

set SET

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

abort examined EXAMINED set SET

### examined

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

## outbound-interface

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

## rpfilter

Performs a reverse path filter test on a packet. If a reply to the packet would be sent via the same interface that the packet arrived on, the packet will match.

```
rpfilter invert true|false
```

### invert

This will invert the sense of the match. Instead of matching packets that passed the reverse path filter test, match those that have failed it.

```
invert true|false
```

### Default value

false

### action

The action performed by this rule.

```
action STANDARD chain <leafref> dscp DSCP reject REJECT \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu \
  tos <0x0-0xff> mask <0x0-0xff>
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <leafref>
```

### dscp

Alters the value of the DSCP bits within the tos header of the IPv4 packet.

```
dscp DSCP
```

| DSCP values | Description   |
|-------------|---|
| <uint8>     | A differentiated services code point (DSCP) marking within the IP header. |
| af11        | AF11 (assured forwarding) class (10).                                     |
| af12        | AF12 (assured forwarding) class (12).                                     |
| af13        | AF13 (assured forwarding) class (14).                                     |
| af21        | AF21 (assured forwarding) class (18).                                     |
| af22        | AF22 (assured forwarding) class (20).                                     |
| af23        | AF23 (assured forwarding) class (22).                                     |
| af31        | AF31 (assured forwarding) class (26).                                     |
| af32        | AF32 (assured forwarding) class (28).                                     |
| af33        | AF33 (assured forwarding) class (30).                                     |
| af41        | AF41 (assured forwarding) class (34).                                     |
| af42        | AF42 (assured forwarding) class (36).                                     |
| af43        | AF43 (assured forwarding) class (38).                                     |
| be          | BE (best effort) class (0).   |
| cs0         | CS0 (class selector) class (0).   |
| cs1         | CS1 (class selector) class (8).   |
| cs2         | CS2 (class selector) class (16).  |
| cs3         | CS3 (class selector) class (24).  |
| cs4         | CS4 (class selector) class (32).  |
| cs5         | CS5 (class selector) class (40).  |
| cs6         | CS6 (class selector) class (48).  |
| cs7         | CS7 (class selector) class (56).  |
| ef          | EF (expedited forwarding) class (46).                                     |

## reject

Used to send back an error packet in response to the matched packet.

```
reject REJECT
```

| REJECT values          | Description   |
|------------------------|---|
| icmp-net-unreachable   | Reject with ICMP network unreachable.   |
| icmp-host-unreachable  | Reject with ICMP host unreachable.  |
| icmp-port-unreachable  | Reject with ICMP port unreachable.  |
| icmp-proto-unreachable | Reject with ICMP prototype unreachable.   |
| icmp-net-prohibited    | Reject with ICMP network prohibited.  |
| icmp-host-prohibited   | Reject with ICMP host prohibited.   |
| icmp-admin-prohibited  | Reject with ICMP admin prohibited.  |
| tcp-reset              | Reject with TCP RST packet. Can be used on rules which only match the TCP protocol. |

**connmark**

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \  
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \  
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \  
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

**set-xmark**

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

XOR with this value.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

**save-mark**

Copy the packet mark (nmask) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nmask \& nmask)$  i.e. ctmask defines what bits to clear and nmask what bits of the nmask to XOR into the ctmark. ctmask and nmask default to 0xFFFFFFFF.

```
save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### **nfmask**

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

### **ctmask**

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

### **restore-mark**

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $\text{nfmark} = (\text{nfmark} \& \sim\text{nfmask}) \wedge (\text{ctmark} \& \text{ctmask})$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### **nfmask**

Bits that should be cleared.

```
nfmask <0x0-0xffffffff>
```

### **ctmask**

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

## log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

## level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

## prefix

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

## additional-infos

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

## mark

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

### <0x0-0xffffffff> (mandatory)

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

### clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

**tos**

Alters the value of the tos header of the IPv4 packet.

```
tos <0x0-0xff> mask <0x0-0xff>
```

**<0x0-0xff> (mandatory)**

Bits that should be XORed into the tos.

```
<0x0-0xff>
```

**mask**

Zero the bits given by this mask in the tos.

```
mask <0x0-0xff>
```

**counters (state only)**

The counters of this rule.

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle chain <string> rule <uint64> ↵  
↪counters packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 mangle chain <string> rule <uint64> ↵  
↪counters bytes
```

## ipv4 raw

---

**Note:** requires a Turbo Router Network License.

---

Mainly used to exempt packets from connection tracking.

```
vrouter running config# vrf <vrf> firewall ipv4 raw
```

## prerouting

Packets as soon as they come in.

```
vrouter running config# vrf <vrf> firewall ipv4 raw prerouting
```

## policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv4 raw prerouting
vrouter running prerouting# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## Default value

accept

## packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 raw prerouting packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 raw prerouting bytes
```

**rule**

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv4 raw prerouting
vrouter running prerouting# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... source \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... ipv4 [not] fragment \
... icmp-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
...   status [not] VALUE \
...   state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
...   rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... inbound-interface [not] <string> \
... rpfILTER invert true|false \
... action STANDARD chain <leafref> notrack \
```

(continues on next page)

(continued from previous page)

```

...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### description

A comment to describe the rule.

```
description <string>
```

### protocol

Match the protocol.

```
protocol [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| icmp         | ICMP protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

## destination

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

## address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C/D>     | IPv4 prefix: address and CIDR mask.  |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE                            | Description  |
|----------------------------------|--|
| val-<br>ues<br><br><domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>                        | v4 address.  |
| <A.B.C.D/M>                      | prefix: address and CIDR mask.   |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**ipv4**

Match the fragment.

```
ipv4 [not] fragment
```

**not**

Invert the match.

```
not
```

**fragment (mandatory)**

Match if the packet is a fragment.

```
fragment
```

**icmp-type**

Match the packet ICMP type.

icmp-type [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The ICMP type to match.

VALUE

| VALUE values              | Description                |
|---------------------------|----------------------------|
| any                       | Any ICMP type.             |
| echo-request              | Echo request.              |
| echo-reply                | Echo reply.                |
| destination-unreachable   | Destination unreachable.   |
| network-unreachable       | Network unreachable.       |
| host-unreachable          | Host unreachable.          |
| protocol-unreachable      | Protocol unreachable.      |
| port-unreachable          | Port unreachable.          |
| fragmentation-needed      | Fragmentation needed.      |
| source-route-failed       | Source route failed.       |
| network-unknown           | Network unknown.           |
| host-unknown              | Host unknown.              |
| network-prohibited        | Network prohibited.        |
| host-prohibited           | Host prohibited.           |
| TOS-network-unreachable   | TOS network unreachable.   |
| TOS-host-unreachable      | TOS host unreachable.      |
| communication-prohibited  | Communication prohibited.  |
| host-precedence-violation | Host precedence violation. |
| precedence-cutoff         | Precedence cutoff.         |
| source-quench             | Source quench.             |
| redirect                  | Redirect.                  |
| network-redirect          | Network redirect.          |

continues on next page

Table 12 – continued from previous page

| VALUE values               | Description                        |
|----------------------------|------------------------------------|
| host-redirect              | Host redirect.                     |
| TOS-network-redirect       | TOS network redirect.              |
| TOS-host-redirect          | TOS host redirect.                 |
| router-advertisement       | Router advertisement.              |
| router-solicitation        | Router solicitation.               |
| ttl-exceeded               | TTL exceeded.                      |
| ttl-zero-during-transit    | Time to Live exceeded in Transit.  |
| ttl-zero-during-reassembly | Fragment Reassembly Time Exceeded. |
| parameter-problem          | Parameter problem.                 |
| ip-header-bad              | Bad IP header.                     |
| required-option-missing    | Missing a Required Option.         |
| timestamp-request          | Timestamp request.                 |
| timestamp-reply            | Timestamp reply.                   |
| address-mask-request       | Address mask request.              |
| address-mask-reply         | Address mask reply.                |

**tcp-flags**

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

**not**

Invert the match.

```
not
```

**set**

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

## examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

## contrack

Match contrack information.

```
contrack \  
  status [not] VALUE \  
  state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \  
shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \  
forward-tsn \  
    data examined EXAMINED set SET \  
    abort examined EXAMINED set SET \  
    shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

abort examined EXAMINED set SET

### examined

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

### rpfilter

Performs a reverse path filter test on a packet. If a reply to the packet would be sent via the same interface that the packet arrived on, the packet will match.

```
rpfilter invert true|false
```

### invert

This will invert the sense of the match. Instead of matching packets that passed the reverse path filter test, match those that have failed it.

```
invert true|false
```

### Default value

false

### action

The action performed by this rule.

```
action STANDARD chain <leafref> notrack \
    connmark \
        set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
        save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
        restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
    mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## chain

Jump to the user chain by this name.

```
chain <leafref>
```

## notrack

Disables connection tracking for this packet.

```
notrack
```

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \  
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \  
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \  
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmask.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

## <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmark \& nfmask)$  i.e. ctmask defines what bits to clear and nfmask what bits of the nfmark to XOR into the ctmark. ctmask and nfmask default to 0xFFFFFFFF.

```
save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmask

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

## ctmask

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

## restore-mark

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $nfmark = (nfmark \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### **nfmask**

Bits that should be cleared.

```
nfmask <0x0-0xffffffff>
```

### **ctmask**

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### **log**

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### **level**

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

**prefix**

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

## set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

## clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

## counters (state only)

The counters of this rule.

## packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 raw prerouting rule <uint64> counters.  
↳packets
```

## bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 raw prerouting rule <uint64> counters bytes
```

## output

Locally-generated packets before routing.

```
vrouter running config# vrf <vrf> firewall ipv4 raw output
```

## policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv4 raw output
vrouter running output# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## Default value

accept

## packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 raw output packets
```

## bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 raw output bytes
```

## rule

A rule to perform an action on matching packets.

```

vrouter running config# vrf <vrf> firewall ipv4 raw output
vrouter running output# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... source \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... ipv4 [not] fragment \
... icmp-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
...   status [not] VALUE \
...   state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
...   rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... outbound-interface [not] <string> \
... action STANDARD chain <leafref> notrack \
...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### **description**

A comment to describe the rule.

```
description <string>
```

### **protocol**

Match the protocol.

```
protocol [not] VALUE
```

### **not**

Invert the match.

```
not
```

### **VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| icmp         | ICMP protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

## destination

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

## address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | prefix: address and CIDR mask.   |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | prefix: address and CIDR mask.   |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**ipv4**

Match the fragment.

```
ipv4 [not] fragment
```

**not**

Invert the match.

```
not
```

**fragment (mandatory)**

Match if the packet is a fragment.

```
fragment
```

**icmp-type**

Match the packet ICMP type.

icmp-type [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The ICMP type to match.

VALUE

| VALUE values              | Description                |
|---------------------------|----------------------------|
| any                       | Any ICMP type.             |
| echo-request              | Echo request.              |
| echo-reply                | Echo reply.                |
| destination-unreachable   | Destination unreachable.   |
| network-unreachable       | Network unreachable.       |
| host-unreachable          | Host unreachable.          |
| protocol-unreachable      | Protocol unreachable.      |
| port-unreachable          | Port unreachable.          |
| fragmentation-needed      | Fragmentation needed.      |
| source-route-failed       | Source route failed.       |
| network-unknown           | Network unknown.           |
| host-unknown              | Host unknown.              |
| network-prohibited        | Network prohibited.        |
| host-prohibited           | Host prohibited.           |
| TOS-network-unreachable   | TOS network unreachable.   |
| TOS-host-unreachable      | TOS host unreachable.      |
| communication-prohibited  | Communication prohibited.  |
| host-precedence-violation | Host precedence violation. |
| precedence-cutoff         | Precedence cutoff.         |
| source-quench             | Source quench.             |
| redirect                  | Redirect.                  |
| network-redirect          | Network redirect.          |

continues on next page

Table 13 – continued from previous page

| VALUE values               | Description                        |
|----------------------------|------------------------------------|
| host-redirect              | Host redirect.                     |
| TOS-network-redirect       | TOS network redirect.              |
| TOS-host-redirect          | TOS host redirect.                 |
| router-advertisement       | Router advertisement.              |
| router-solicitation        | Router solicitation.               |
| ttl-exceeded               | TTL exceeded.                      |
| ttl-zero-during-transit    | Time to Live exceeded in Transit.  |
| ttl-zero-during-reassembly | Fragment Reassembly Time Exceeded. |
| parameter-problem          | Parameter problem.                 |
| ip-header-bad              | Bad IP header.                     |
| required-option-missing    | Missing a Required Option.         |
| timestamp-request          | Timestamp request.                 |
| timestamp-reply            | Timestamp reply.                   |
| address-mask-request       | Address mask request.              |
| address-mask-reply         | Address mask reply.                |

**tcp-flags**

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

**not**

Invert the match.

```
not
```

**set**

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

**contrack**

Match contrack information.

```
contrack \  
  status [not] VALUE \  
  state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \  
shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \  
forward-tsn \  
    data examined EXAMINED set SET \  
    abort examined EXAMINED set SET \  
    shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

**abort**

ABORT chunk.

abort examined EXAMINED set SET

**examined**

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

**set**

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

**shutdown-complete**

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### outbound-interface

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

### action

The action performed by this rule.

```
action STANDARD chain <leafref> notrack \
    connmark \
        set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
        save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
        restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
    mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <leafref>
```

### notrack

Disables connection tracking for this packet.

```
notrack
```

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

## <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nmask) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nmask \& nmask)$  i.e. ctmask defines what bits to clear and nmask what bits of the nmask to XOR into the ctmark. ctmask and nmask default to 0xFFFFFFFF.

```
save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### **nfmask**

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

### **ctmask**

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

### **restore-mark**

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $\text{nfmark} = (\text{nfmark} \& \sim\text{nfmask}) \wedge (\text{ctmark} \& \text{ctmask})$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### **nfmask**

Bits that should be cleared.

```
nfmask <0x0-0xffffffff>
```

### **ctmask**

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

### prefix

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

### additional-infos

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

## mark

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

### <0x0-0xffffffff> (mandatory)

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

### clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

### counters (state only)

The counters of this rule.

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 raw output rule <uint64> counters packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 raw output rule <uint64> counters bytes
```

### chain

User chain.

```
vrouter running config# vrf <vrf> firewall ipv4 raw chain <string>
```

|          |                      |
|----------|----------------------|
| <string> | The user chain name. |
|----------|----------------------|

### policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv4 raw chain <string>
vrouter running chain <string># policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### Default value

accept

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 raw chain <string> packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 raw chain <string> bytes
```

**rule**

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv4 raw chain <string>
vrouter running chain <string># rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... source \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... ipv4 [not] fragment \
... icmp-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
... status [not] VALUE \
... state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
... rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
```

(continues on next page)

(continued from previous page)

```

... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... inbound-interface [not] <string> \
... outbound-interface [not] <string> \
... rpfiler invert true|false \
... action STANDARD chain <leafref> dscp DSCP reject REJECT \
...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu \
...   tos <0x0-0xff> mask <0x0-0xff>

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### description

A comment to describe the rule.

```
description <string>
```

### protocol

Match the protocol.

```
protocol [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

The protocol to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| icmp         | ICMP protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |

### destination

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

### address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C/D>     | IPv4 prefix: address and CIDR mask.  |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D>     | IPv4 address.  |
| <A.B.C.D/M>   | prefix: address and CIDR mask.   |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**ipv4**

Match the fragment.

```
ipv4 [not] fragment
```

**not**

Invert the match.

```
not
```

**fragment (mandatory)**

Match if the packet is a fragment.

```
fragment
```

**icmp-type**

Match the packet ICMP type.

icmp-type [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The ICMP type to match.

VALUE

| VALUE values              | Description                |
|---------------------------|----------------------------|
| any                       | Any ICMP type.             |
| echo-request              | Echo request.              |
| echo-reply                | Echo reply.                |
| destination-unreachable   | Destination unreachable.   |
| network-unreachable       | Network unreachable.       |
| host-unreachable          | Host unreachable.          |
| protocol-unreachable      | Protocol unreachable.      |
| port-unreachable          | Port unreachable.          |
| fragmentation-needed      | Fragmentation needed.      |
| source-route-failed       | Source route failed.       |
| network-unknown           | Network unknown.           |
| host-unknown              | Host unknown.              |
| network-prohibited        | Network prohibited.        |
| host-prohibited           | Host prohibited.           |
| TOS-network-unreachable   | TOS network unreachable.   |
| TOS-host-unreachable      | TOS host unreachable.      |
| communication-prohibited  | Communication prohibited.  |
| host-precedence-violation | Host precedence violation. |
| precedence-cutoff         | Precedence cutoff.         |
| source-quench             | Source quench.             |
| redirect                  | Redirect.                  |
| network-redirect          | Network redirect.          |

continues on next page

Table 14 – continued from previous page

| VALUE values               | Description                        |
|----------------------------|------------------------------------|
| host-redirect              | Host redirect.                     |
| TOS-network-redirect       | TOS network redirect.              |
| TOS-host-redirect          | TOS host redirect.                 |
| router-advertisement       | Router advertisement.              |
| router-solicitation        | Router solicitation.               |
| ttl-exceeded               | TTL exceeded.                      |
| ttl-zero-during-transit    | Time to Live exceeded in Transit.  |
| ttl-zero-during-reassembly | Fragment Reassembly Time Exceeded. |
| parameter-problem          | Parameter problem.                 |
| ip-header-bad              | Bad IP header.                     |
| required-option-missing    | Missing a Required Option.         |
| timestamp-request          | Timestamp request.                 |
| timestamp-reply            | Timestamp reply.                   |
| address-mask-request       | Address mask request.              |
| address-mask-reply         | Address mask reply.                |

**tcp-flags**

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

**not**

Invert the match.

```
not
```

**set**

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
  status [not] VALUE \  
  state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

## sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \
  shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \
  forward-tsn \
  data examined EXAMINED set SET \
  abort examined EXAMINED set SET \
  shutdown-complete examined EXAMINED set SET
```

### not

Invert the match.

```
not
```

### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

abort examined EXAMINED set SET

### examined

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

## examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

## set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

## inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

## not

Invert the match.

```
not
```

## <string> (mandatory)

The interface to match.

```
<string>
```

## outbound-interface

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

## rpfilter

Performs a reverse path filter test on a packet. If a reply to the packet would be sent via the same interface that the packet arrived on, the packet will match.

```
rpfilter invert true|false
```

### invert

This will invert the sense of the match. Instead of matching packets that passed the reverse path filter test, match those that have failed it.

```
invert true|false
```

### Default value

false

### action

The action performed by this rule.

```
action STANDARD chain <leafref> dscp DSCP reject REJECT \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu \
  tos <0x0-0xff> mask <0x0-0xff>
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <leafref>
```

### dscp

Alters the value of the DSCP bits within the tos header of the IPv4 packet.

```
dscp DSCP
```

| DSCP values | Description   |
|-------------|---|
| <uint8>     | A differentiated services code point (DSCP) marking within the IP header. |
| af11        | AF11 (assured forwarding) class (10).                                     |
| af12        | AF12 (assured forwarding) class (12).                                     |
| af13        | AF13 (assured forwarding) class (14).                                     |
| af21        | AF21 (assured forwarding) class (18).                                     |
| af22        | AF22 (assured forwarding) class (20).                                     |
| af23        | AF23 (assured forwarding) class (22).                                     |
| af31        | AF31 (assured forwarding) class (26).                                     |
| af32        | AF32 (assured forwarding) class (28).                                     |
| af33        | AF33 (assured forwarding) class (30).                                     |
| af41        | AF41 (assured forwarding) class (34).                                     |
| af42        | AF42 (assured forwarding) class (36).                                     |
| af43        | AF43 (assured forwarding) class (38).                                     |
| be          | BE (best effort) class (0).   |
| cs0         | CS0 (class selector) class (0).   |
| cs1         | CS1 (class selector) class (8).   |
| cs2         | CS2 (class selector) class (16).  |
| cs3         | CS3 (class selector) class (24).  |
| cs4         | CS4 (class selector) class (32).  |
| cs5         | CS5 (class selector) class (40).  |
| cs6         | CS6 (class selector) class (48).  |
| cs7         | CS7 (class selector) class (56).  |
| ef          | EF (expedited forwarding) class (46).                                     |

## reject

Used to send back an error packet in response to the matched packet.

```
reject REJECT
```

| REJECT values          | Description   |
|------------------------|---|
| icmp-net-unreachable   | Reject with ICMP network unreachable.   |
| icmp-host-unreachable  | Reject with ICMP host unreachable.  |
| icmp-port-unreachable  | Reject with ICMP port unreachable.  |
| icmp-proto-unreachable | Reject with ICMP prototype unreachable.   |
| icmp-net-prohibited    | Reject with ICMP network prohibited.  |
| icmp-host-prohibited   | Reject with ICMP host prohibited.   |
| icmp-admin-prohibited  | Reject with ICMP admin prohibited.  |
| tcp-reset              | Reject with TCP RST packet. Can be used on rules which only match the TCP protocol. |

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \  
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \  
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \  
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

## <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nmask) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nmask \& nmask)$  i.e. ctmask defines what bits to clear and nmask what bits of the nmask to XOR into the ctmark. ctmask and nmask default to 0xFFFFFFFF.

```
save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### **nfmask**

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

### **ctmask**

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

### **restore-mark**

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $\text{nfmark} = (\text{nfmark} \& \sim\text{nfmask}) \wedge (\text{ctmark} \& \text{ctmask})$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### **nfmask**

Bits that should be cleared.

```
nfmask <0x0-0xffffffff>
```

### **ctmask**

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

## log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

## level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

## prefix

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

## additional-infos

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

## mark

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

### <0x0-0xffffffff> (mandatory)

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

### clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

**tos**

Alters the value of the tos header of the IPv4 packet.

```
tos <0x0-0xff> mask <0x0-0xff>
```

**<0x0-0xff> (mandatory)**

Bits that should be XORed into the tos.

```
<0x0-0xff>
```

**mask**

Zero the bits given by this mask in the tos.

```
mask <0x0-0xff>
```

**counters (state only)**

The counters of this rule.

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv4 raw chain <string> rule <uint64> counters_
↳packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv4 raw chain <string> rule <uint64> counters_
↳bytes
```

## ipv4 address group

---

**Note:** requires a Turbo Router Network License.

---

**Attention:**

**Deprecated since:** 2021-01-04

**Obsolete in release:** 21q3

**Description:** This configuration has been moved outside of the firewall configuration to be usable in other contexts.

**Replacement:** / vrf group ipv4 address-group

Address group.

```
vrouter running config# vrf <vrf> firewall ipv4 address-group <string>
```

|          |                            |
|----------|----------------------------|
| <string> | Name of the address group. |
|----------|----------------------------|

## address (deprecated)

List of addresses of the group.

```
vrouter running config# vrf <vrf> firewall ipv4 address-group <string>
vrouter running address-group <string># address ADDRESS
```

|          |   |
|----------|---|
| AD-DRESS | An IPv4 address without a zone index. This type, derived from ipv4-address, may be used in situations where the zone is known from the context and hence no zone index is needed. |
|----------|---|

## ipv4 network group

---

**Note:** requires a Turbo Router Network License.

---

**Attention:**

**Deprecated since:** 2021-01-04

**Obsolete in release:** 21q3

**Description:** This configuration has been moved outside of the firewall configuration to be usable in other contexts.

**Replacement:** / vrf group ipv4 network-group

Network group.

```
vrouter running config# vrf <vrf> firewall ipv4 network-group <string>
```

|          |                            |
|----------|----------------------------|
| <string> | Name of the network group. |
|----------|----------------------------|

### network (deprecated)

List of networks of the group.

```
vrouter running config# vrf <vrf> firewall ipv4 network-group <string>
vrouter running network-group <string># network NETWORK
```

|         |  |
|---------|--|
| NETWORK | An IPv4 prefix: address and CIDR mask. |
|---------|--|

### ipv6 filter

Default table.

```
vrouter running config# vrf <vrf> firewall ipv6 filter
```

### input

Packets destined to local sockets.

```
vrouter running config# vrf <vrf> firewall ipv6 filter input
```

### policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv6 filter input
vrouter running input# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### Default value

accept

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 filter input packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 filter input bytes
```

### rule

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv6 filter input
vrouter running input# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... source \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... icmpv6-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
```

(continues on next page)

(continued from previous page)

```

...  conntrack \
...    status [not] VALUE \
...    state [not] VALUE \
...  connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
...  limit burst <uint32> \
...    rate <uint32> UNIT \
...  dscp [not] VALUE \
...  tos [not] <0x0-0xff> mask <0x0-0xff> \
...  mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
...  sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
...  shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
...  asconf-ack forward-tsn \
...    data examined EXAMINED set SET \
...    abort examined EXAMINED set SET \
...    shutdown-complete examined EXAMINED set SET \
...  inbound-interface [not] <string> \
...  action STANDARD chain <string> reject REJECT \
...    connmark \
...      set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...      save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...      restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...    log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...    mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...    tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### description

A comment to describe the rule.

```
description <string>
```

### protocol

Match the protocol.

```
protocol [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The protocol to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| ipv6-icmp    | ICMPv6 protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |

**destination**

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

### address

Match on destination address.

```
address [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

The address to match.

```
VALUE
```

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on destination port.

```
port [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on destination port range (syntax: port[,port|port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

### VALUE (mandatory)

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

### group

Matches a set of addresses or networks.

```
group [not] <string>
```

### not

Not match-set.

```
not
```

### <string> (mandatory)

The name of the group.

```
<string>
```

### source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

**address**

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The address to match.

```
VALUE
```

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on source port.

```
port [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

### VALUE (mandatory)

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

### group

Matches a set of addresses or networks.

```
group [not] <string>
```

### not

Not match-set.

```
not
```

### <string> (mandatory)

The name of the group.

```
<string>
```

### icmpv6-type

Match the packet ICMP type.

```
icmpv6-type [not] VALUE
```

**not**

Invert the match.

|     |
|-----|
| not |
|-----|

**VALUE (mandatory)**

The ICMP type to match.

|       |
|-------|
| VALUE |
|-------|

| VALUE values               | Description   |
|----------------------------|---|
| echo-request               | Echo request.   |
| echo-reply                 | Echo reply.   |
| destination-unreachable    | Destination unreachable.                                    |
| address-unreachable        | Address unreachable.  |
| port-unreachable           | Port unreachable.   |
| no-route                   | No route to destination.                                    |
| reject-route               | Reject route to destination.                                |
| communication-prohibited   | Communication with destination administratively prohibited. |
| beyond-scope               | Beyond scope of source address.                             |
| packet-too-big             | Packet too big.   |
| failed-policy              | Source address failed ingress/egress policy.                |
| ttl-exceeded               | TTL exceeded.   |
| ttl-zero-during-transit    | Hop limit exceeded in transit.                              |
| ttl-zero-during-reassembly | Fragment reassembly time exceeded.                          |
| parameter-problem          | Parameter problem.  |
| bad-header                 | Erroneous header field encountered.                         |
| unknown-header-type        | Unrecognized Next Header type encountered.                  |
| unknown-option             | Unrecognized IPv6 option encountered.                       |
| router-solicitation        | Router solicitation.  |
| router-advertisement       | Router advertisement.                                       |
| neighbor-solicitation      | Neighbor solicitation.                                      |
| neighbor-advertisement     | Neighbor advertisement.                                     |
| redirect                   | Redirect message.   |

**tcp-flags**

Match the packet TCP flags.

tcp-flags [not] set SET examined EXAMINED

**not**

Invert the match.

not

**set**

Set flags.

set SET

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

**examined**

Examined flags.

examined EXAMINED

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

### status

Match the connection status.

```
status [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE

The contrack status to match.

```
VALUE
```

| VALUE values | Description  |
|--------------|--|
| none         | No status.   |
| expected     | This is an expected connection (i.e. a contrack helper set it up). |
| seen_reply   | Contrack has seen packets in both directions.                      |
| assured      | Contrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.          |

**state**

Match the packet state regarding contrack.

```
state [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE**

The packet states to match.

```
VALUE
```

| VALUE val-ues | Description  |
|---------------|--|
| invalid       | Packet is associated with no known connection.   |
| new           | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| estab-lished  | Packet is associated with a connection which has seen packets in both directions.  |
| related       | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked     | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat          | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat          | A virtual state, matching if the original destination differs from the reply source.   |

**connmark**

Matches the mark field associated with a connection.

```
connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

```
dscp [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The DSCP value to match.

|       |
|-------|
| VALUE |
|-------|

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

|                                      |
|--------------------------------------|
| tos [not] <0x0-0xff> mask <0x0-0xff> |
|--------------------------------------|

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**sctp-chunk-types**

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \
  shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \
  forward-tsn \
  data examined EXAMINED set SET \
  abort examined EXAMINED set SET \
  shutdown-complete examined EXAMINED set SET
```

**not**

Invert the match.

```
not
```

**SCOPE (mandatory)**

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

**init**

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

set SET

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

**abort**

ABORT chunk.

abort examined EXAMINED set SET

**examined**

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

**set**

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

```
shutdown-complete examined EXAMINED set SET
```

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

**<string> (mandatory)**

The interface to match.

```
<string>
```

**action**

The action performed by this rule.

```
action STANDARD chain <string> reject REJECT \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

**STANDARD**

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

**chain**

Jump to the user chain by this name.

```
chain <string>
```

## reject

Used to send back an error packet in response to the matched packet.

```
reject REJECT
```

| REJECT values          | Description   |
|------------------------|---|
| icmp6-no-route         | Reject with ICMPv6 no route.  |
| icmp6-adm-prohibited   | Reject with ICMPv6 admin prohibited.  |
| icmp6-addr-unreachable | Reject with ICMPv6 address unreachable.   |
| icmp6-port-unreachable | Reject with ICMPv6 port unreachable.  |
| tcp-reset              | Reject with TCP RST packet. Can be used on rules which only match the TCP protocol. |

## conmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
conmark \
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

## <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmark \& nfmask)$  i.e. ctmask defines what bits to clear and nfmask what bits of the nfmark to XOR into the ctmark. ctmask and nfmask default to 0xFFFFFFFF.

```
save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmask

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

## ctmask

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

## restore-mark

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $nfmark = (nfmark \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### **nfmask**

Bits that should be cleared.

```
nfmask <0x0-0xffffffff>
```

### **ctmask**

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### **log**

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### **level**

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

**prefix**

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

## set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

## clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

## counters (state only)

The counters of this rule.

## packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 filter input rule <uint64> counters packets
```

## bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 filter input rule <uint64> counters bytes
```

### forward

Packets being routed.

```
vrouter running config# vrf <vrf> firewall ipv6 filter forward
```

### policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv6 filter forward
vrouter running forward# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### Default value

accept

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 filter forward packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 filter forward bytes
```

## rule

A rule to perform an action on matching packets.

```

vrouter running config# vrf <vrf> firewall ipv6 filter forward
vrouter running forward# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... source \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... icmpv6-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... contrack \
...   status [not] VALUE \
...   state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
...   rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... inbound-interface [not] <string> \
... outbound-interface [not] <string> \
... action STANDARD chain <string> reject REJECT \
...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### **description**

A comment to describe the rule.

```
description <string>
```

### **protocol**

Match the protocol.

```
protocol [not] VALUE
```

### **not**

Invert the match.

```
not
```

### **VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| ipv6-icmp    | ICMPv6 protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

## destination

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

## address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

|     |
|-----|
| not |
|-----|

**VALUE (mandatory)**

The address to match.

|       |
|-------|
| VALUE |
|-------|

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on source port.

|                  |
|------------------|
| port [not] VALUE |
|------------------|

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**icmpv6-type**

Match the packet ICMP type.

```
icmpv6-type [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The ICMP type to match.

```
VALUE
```

| VALUE values               | Description   |
|----------------------------|---|
| echo-request               | Echo request.   |
| echo-reply                 | Echo reply.   |
| destination-unreachable    | Destination unreachable.                                    |
| address-unreachable        | Address unreachable.  |
| port-unreachable           | Port unreachable.   |
| no-route                   | No route to destination.                                    |
| reject-route               | Reject route to destination.                                |
| communication-prohibited   | Communication with destination administratively prohibited. |
| beyond-scope               | Beyond scope of source address.                             |
| packet-too-big             | Packet too big.   |
| failed-policy              | Source address failed ingress/egress policy.                |
| ttl-exceeded               | TTL exceeded.   |
| ttl-zero-during-transit    | Hop limit exceeded in transit.                              |
| ttl-zero-during-reassembly | Fragment reassembly time exceeded.                          |
| parameter-problem          | Parameter problem.  |
| bad-header                 | Erroneous header field encountered.                         |
| unknown-header-type        | Unrecognized Next Header type encountered.                  |
| unknown-option             | Unrecognized IPv6 option encountered.                       |
| router-solicitation        | Router solicitation.  |
| router-advertisement       | Router advertisement.                                       |
| neighbor-solicitation      | Neighbor solicitation.                                      |
| neighbor-advertisement     | Neighbor advertisement.                                     |
| redirect                   | Redirect message.   |

### tcp-flags

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

### not

Invert the match.

```
not
```

### set

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \  
shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \  
forward-tsn \  
    data examined EXAMINED set SET \  
    abort examined EXAMINED set SET \  
    shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

```
abort examined EXAMINED set SET
```

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

```
shutdown-complete examined EXAMINED set SET
```

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

**outbound-interface**

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

**not**

Invert the match.

```
not
```

**<string> (mandatory)**

The interface to match.

```
<string>
```

**action**

The action performed by this rule.

```
action STANDARD chain <string> reject REJECT \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

**STANDARD**

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## chain

Jump to the user chain by this name.

```
chain <string>
```

## reject

Used to send back an error packet in response to the matched packet.

```
reject REJECT
```

| REJECT values          | Description   |
|------------------------|---|
| icmp6-no-route         | Reject with ICMPv6 no route.  |
| icmp6-adm-prohibited   | Reject with ICMPv6 admin prohibited.  |
| icmp6-addr-unreachable | Reject with ICMPv6 address unreachable.   |
| icmp6-port-unreachable | Reject with ICMPv6 port unreachable.  |
| tcp-reset              | Reject with TCP RST packet. Can be used on rules which only match the TCP protocol. |

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \  
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \  
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \  
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmask.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

### <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nfmask) to the connection mark (ctmask) using the given masks. The new value is determined as follows:  $ctmask = (ctmask \& \sim ctmask) \wedge (nfmask \& nfmask)$  i.e. ctmask defines what bits to clear and nfmask what bits of the nfmask to XOR into the ctmask. ctmask and nfmask default to 0xFFFFFFFF.

```
save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmask

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

## ctmask

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

## restore-mark

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $\text{nfmark} = (\text{nfmark} \& \sim\text{nfmask}) \wedge (\text{ctmark} \& \text{ctmask})$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmask

Bits that should be cleared.

```
nfmask <0x0-0xffffffff>
```

## ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

## log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

## level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

**prefix**

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

**tcpmss**

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

**set-mss**

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

**clamp-mss-to-pmtu**

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

**counters (state only)**

The counters of this rule.

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 filter forward rule <uint64> counters.
↳ packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 filter forward rule <uint64> counters bytes
```

**output**

Locally-generated packets.

```
vrouter running config# vrf <vrf> firewall ipv6 filter output
```

**policy**

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv6 filter output
vrouter running output# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

**Default value**

accept

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 filter output packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 filter output bytes
```

**rule**

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv6 filter output
vrouter running output# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... source \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... icmpv6-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
... status [not] VALUE \
... state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
... rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
```

(continues on next page)

(continued from previous page)

```

...  asconf-ack forward-tsn \
...    data examined EXAMINED set SET \
...    abort examined EXAMINED set SET \
...    shutdown-complete examined EXAMINED set SET \
...  outbound-interface [not] <string> \
...  action STANDARD chain <string> reject REJECT \
...    connmark \
...      set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...      save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...      restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...    log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...    mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...    tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### description

A comment to describe the rule.

```
description <string>
```

### protocol

Match the protocol.

```
protocol [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

The protocol to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| ipv6-icmp    | ICMPv6 protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |

### destination

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

### address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**icmpv6-type**

Match the packet ICMP type.

```
icmpv6-type [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The ICMP type to match.

```
VALUE
```

| VALUE values               | Description   |
|----------------------------|---|
| echo-request               | Echo request.   |
| echo-reply                 | Echo reply.   |
| destination-unreachable    | Destination unreachable.                                    |
| address-unreachable        | Address unreachable.  |
| port-unreachable           | Port unreachable.   |
| no-route                   | No route to destination.                                    |
| reject-route               | Reject route to destination.                                |
| communication-prohibited   | Communication with destination administratively prohibited. |
| beyond-scope               | Beyond scope of source address.                             |
| packet-too-big             | Packet too big.   |
| failed-policy              | Source address failed ingress/egress policy.                |
| ttl-exceeded               | TTL exceeded.   |
| ttl-zero-during-transit    | Hop limit exceeded in transit.                              |
| ttl-zero-during-reassembly | Fragment reassembly time exceeded.                          |
| parameter-problem          | Parameter problem.  |
| bad-header                 | Erroneous header field encountered.                         |
| unknown-header-type        | Unrecognized Next Header type encountered.                  |
| unknown-option             | Unrecognized IPv6 option encountered.                       |
| router-solicitation        | Router solicitation.  |
| router-advertisement       | Router advertisement.                                       |
| neighbor-solicitation      | Neighbor solicitation.                                      |
| neighbor-advertisement     | Neighbor advertisement.                                     |
| redirect                   | Redirect message.   |

### tcp-flags

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

### not

Invert the match.

```
not
```

### set

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \  
shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \  
forward-tsn \  
    data examined EXAMINED set SET \  
    abort examined EXAMINED set SET \  
    shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

asconf-ack

**forward-tsn**

FORWARD TSN chunk.

forward-tsn

**data**

DATA chunk.

data examined EXAMINED set SET

**examined**

Examined flags.

examined EXAMINED

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

set SET

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

```
abort examined EXAMINED set SET
```

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

```
shutdown-complete examined EXAMINED set SET
```

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### outbound-interface

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

### action

The action performed by this rule.

```
action STANDARD chain <string> reject REJECT \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <string>
```

### reject

Used to send back an error packet in response to the matched packet.

```
reject REJECT
```

| REJECT values          | Description   |
|------------------------|---|
| icmp6-no-route         | Reject with ICMPv6 no route.  |
| icmp6-adm-prohibited   | Reject with ICMPv6 admin prohibited.  |
| icmp6-addr-unreachable | Reject with ICMPv6 address unreachable.   |
| icmp6-port-unreachable | Reject with ICMPv6 port unreachable.  |
| tcp-reset              | Reject with TCP RST packet. Can be used on rules which only match the TCP protocol. |

### connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \  
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \  
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \  
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

### <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

### mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nfmak) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmak \& nfmask)$  i.e. ctmask defines what bits to clear and nfmak what bits of the nfmak to XOR into the ctmark. ctmask and nfmak default to 0xFFFFFFFF.

```
save-mark nfmak <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmak

Bits that should be XORed into the connection mark.

```
nfmak <0x0-0xffffffff>
```

## ctmask

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

## restore-mark

Copy the connection mark (ctmark) to the packet mark (nfmak) using the given masks. The new ctmark value is determined as follows:  $nfmak = (nfmak \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmak defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmak default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmak <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmak

Bits that should be cleared.

```
nfmak <0x0-0xffffffff>
```

### ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

### prefix

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

**tcpmss**

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

### clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

### counters (state only)

The counters of this rule.

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 filter output rule <uint64> counters_
↳packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 filter output rule <uint64> counters bytes
```

### chain

User chain.

```
vrouter running config# vrf <vrf> firewall ipv6 filter chain <string>
```

|          |                      |
|----------|----------------------|
| <string> | The user chain name. |
|----------|----------------------|

## policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv6 filter chain <string>
vrouter running chain <string># policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### Default value

accept

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 filter chain <string> packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 filter chain <string> bytes
```

## rule

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv6 filter chain <string>
vrouter running chain <string># rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
```

(continues on next page)

(continued from previous page)

```

... source \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
...   icmpv6-type [not] VALUE \
...   tcp-flags [not] set SET examined EXAMINED \
...   conntrack \
...     status [not] VALUE \
...     state [not] VALUE \
...   connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   limit burst <uint32> \
...     rate <uint32> UNIT \
...   dscp [not] VALUE \
...   tos [not] <0x0-0xff> mask <0x0-0xff> \
...   mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
...   shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
...   asconf-ack forward-tsn \
...     data examined EXAMINED set SET \
...     abort examined EXAMINED set SET \
...     shutdown-complete examined EXAMINED set SET \
...   inbound-interface [not] <string> \
...   outbound-interface [not] <string> \
...   rpfilter invert true|false \
...   action STANDARD chain <string> reject REJECT \
...     connmark \
...       set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...       save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...       restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

**description**

A comment to describe the rule.

```
description <string>
```

**protocol**

Match the protocol.

```
protocol [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| ipv6-icmp    | ICMPv6 protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

## destination

Match on destination fields.

```
destination \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on destination address.

```
address [not] VALUE
```

## not

Invert the match.

```
not
```

## VALUE (mandatory)

The address to match.

```
VALUE
```

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | An IPv6 address.   |
| <X:X::X/X>    | An IPv6 prefix: address and CIDR mask.   |

### port

Match on destination port.

port [not] VALUE

### not

Invert the match.

not

### VALUE (mandatory)

The port to match.

VALUE

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

## port-range

Match on destination port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

### not

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**source**

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

**address**

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The address to match.

```
VALUE
```

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | An IPv6 address.   |
| <X:X::X/X>    | An IPv6 prefix: address and CIDR mask.   |

### port

Match on source port.

port [not] VALUE

### not

Invert the match.

not

### VALUE (mandatory)

The port to match.

VALUE

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

### port-range

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

### group

Matches a set of addresses or networks.

```
group [not] <string>
```

### not

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**icmpv6-type**

Match the packet ICMP type.

```
icmpv6-type [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The ICMP type to match.

```
VALUE
```

| VALUE values               | Description   |
|----------------------------|---|
| echo-request               | Echo request.   |
| echo-reply                 | Echo reply.   |
| destination-unreachable    | Destination unreachable.                                    |
| address-unreachable        | Address unreachable.  |
| port-unreachable           | Port unreachable.   |
| no-route                   | No route to destination.                                    |
| reject-route               | Reject route to destination.                                |
| communication-prohibited   | Communication with destination administratively prohibited. |
| beyond-scope               | Beyond scope of source address.                             |
| packet-too-big             | Packet too big.   |
| failed-policy              | Source address failed ingress/egress policy.                |
| ttl-exceeded               | TTL exceeded.   |
| ttl-zero-during-transit    | Hop limit exceeded in transit.                              |
| ttl-zero-during-reassembly | Fragment reassembly time exceeded.                          |
| parameter-problem          | Parameter problem.  |
| bad-header                 | Erroneous header field encountered.                         |
| unknown-header-type        | Unrecognized Next Header type encountered.                  |
| unknown-option             | Unrecognized IPv6 option encountered.                       |
| router-solicitation        | Router solicitation.  |
| router-advertisement       | Router advertisement.                                       |
| neighbor-solicitation      | Neighbor solicitation.                                      |
| neighbor-advertisement     | Neighbor advertisement.                                     |
| redirect                   | Redirect message.   |

### tcp-flags

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

### not

Invert the match.

```
not
```

### set

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \  
shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \  
forward-tsn \  
    data examined EXAMINED set SET \  
    abort examined EXAMINED set SET \  
    shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

abort examined EXAMINED set SET

### examined

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

## outbound-interface

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

## rpfilter

Performs a reverse path filter test on a packet. If a reply to the packet would be sent via the same interface that the packet arrived on, the packet will match.

```
rpfilter invert true|false
```

### invert

This will invert the sense of the match. Instead of matching packets that passed the reverse path filter test, match those that have failed it.

```
invert true|false
```

### Default value

false

### action

The action performed by this rule.

```
action STANDARD chain <string> reject REJECT \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <string>
```

### reject

Used to send back an error packet in response to the matched packet.

```
reject REJECT
```

| REJECT values          | Description   |
|------------------------|---|
| icmp6-no-route         | Reject with ICMPv6 no route.  |
| icmp6-adm-prohibited   | Reject with ICMPv6 admin prohibited.  |
| icmp6-addr-unreachable | Reject with ICMPv6 address unreachable.   |
| icmp6-port-unreachable | Reject with ICMPv6 port unreachable.  |
| tcp-reset              | Reject with TCP RST packet. Can be used on rules which only match the TCP protocol. |

### connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \  
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \  
  save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \  
  restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

### <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

### mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nfmak) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmak \& nfmask)$  i.e. ctmask defines what bits to clear and nfmak what bits of the nfmak to XOR into the ctmark. ctmask and nfmak default to 0xFFFFFFFF.

```
save-mark nfmak <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmak

Bits that should be XORed into the connection mark.

```
nfmak <0x0-0xffffffff>
```

## ctmask

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

## restore-mark

Copy the connection mark (ctmark) to the packet mark (nfmak) using the given masks. The new ctmark value is determined as follows:  $nfmak = (nfmak \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmak defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmak default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmak <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmak

Bits that should be cleared.

```
nfmak <0x0-0xffffffff>
```

### ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

### prefix

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

**tcpmss**

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

### clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

### counters (state only)

The counters of this rule.

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 filter chain <string> rule <uint64> ↵  
↪counters packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 filter chain <string> rule <uint64> ↵  
↪counters bytes
```

### ipv6 mangle

Packet alteration table.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle
```

### prerouting

Altering packets as soon as they come in.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle prerouting
```

### policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle prerouting
vrouter running prerouting# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### Default value

accept

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle prerouting packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle prerouting bytes
```

## rule

A rule to perform an action on matching packets.

```

vrouter running config# vrf <vrf> firewall ipv6 mangle prerouting
vrouter running prerouting# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... source \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... icmpv6-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... contrack \
...   status [not] VALUE \
...   state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
...   rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... inbound-interface [not] <string> \
... rpfILTER invert true|false \
... action STANDARD chain <string> \
...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-Infos ADDITIONAL-Infos \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### **description**

A comment to describe the rule.

```
description <string>
```

### **protocol**

Match the protocol.

```
protocol [not] VALUE
```

### **not**

Invert the match.

```
not
```

### **VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| ipv6-icmp    | ICMPv6 protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |

### destination

Match on destination fields.

```
destination \
    address [not] VALUE \
    port [not] VALUE \
    port-range [not] VALUE \
    group [not] <string>
```

### address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE                            | Description  |
|----------------------------------|--|
| val-<br>ues<br><br><domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>                       | IPv6 address.  |
| <X:X::X/X>                       | IPv6 prefix: address and CIDR mask.  |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**icmpv6-type**

Match the packet ICMP type.

```
icmpv6-type [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The ICMP type to match.

```
VALUE
```

| VALUE values               | Description   |
|----------------------------|---|
| echo-request               | Echo request.   |
| echo-reply                 | Echo reply.   |
| destination-unreachable    | Destination unreachable.                                    |
| address-unreachable        | Address unreachable.  |
| port-unreachable           | Port unreachable.   |
| no-route                   | No route to destination.                                    |
| reject-route               | Reject route to destination.                                |
| communication-prohibited   | Communication with destination administratively prohibited. |
| beyond-scope               | Beyond scope of source address.                             |
| packet-too-big             | Packet too big.   |
| failed-policy              | Source address failed ingress/egress policy.                |
| ttl-exceeded               | TTL exceeded.   |
| ttl-zero-during-transit    | Hop limit exceeded in transit.                              |
| ttl-zero-during-reassembly | Fragment reassembly time exceeded.                          |
| parameter-problem          | Parameter problem.  |
| bad-header                 | Erroneous header field encountered.                         |
| unknown-header-type        | Unrecognized Next Header type encountered.                  |
| unknown-option             | Unrecognized IPv6 option encountered.                       |
| router-solicitation        | Router solicitation.  |
| router-advertisement       | Router advertisement.                                       |
| neighbor-solicitation      | Neighbor solicitation.                                      |
| neighbor-advertisement     | Neighbor advertisement.                                     |
| redirect                   | Redirect message.   |

### tcp-flags

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

### not

Invert the match.

```
not
```

### set

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

## sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \
  shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \
  forward-tsn \
  data examined EXAMINED set SET \
  abort examined EXAMINED set SET \
  shutdown-complete examined EXAMINED set SET
```

### not

Invert the match.

```
not
```

### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

asconf-ack

**forward-tsn**

FORWARD TSN chunk.

forward-tsn

**data**

DATA chunk.

data examined EXAMINED set SET

**examined**

Examined flags.

examined EXAMINED

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

set SET

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

abort examined EXAMINED set SET

### examined

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

## examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

## set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

## inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

## not

Invert the match.

```
not
```

## <string> (mandatory)

The interface to match.

```
<string>
```

## rpfilter

Performs a reverse path filter test on a packet. If a reply to the packet would be sent via the same interface that the packet arrived on, the packet will match.

```
rpfilter invert true|false
```

## invert

This will invert the sense of the match. Instead of matching packets that passed the reverse path filter test, match those that have failed it.

```
invert true|false
```

## Default value

false

## action

The action performed by this rule.

```
action STANDARD chain <string> \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

## STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## chain

Jump to the user chain by this name.

```
chain <string>
```

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \  
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \  
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \  
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmask.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

## <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nfmak) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmak \& nfmask)$  i.e. ctmask defines what bits to clear and nfmak what bits of the nfmak to XOR into the ctmark. ctmask and nfmak default to 0xFFFFFFFF.

```
save-mark nfmak <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmak

Bits that should be XORed into the connection mark.

```
nfmak <0x0-0xffffffff>
```

## ctmask

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

## restore-mark

Copy the connection mark (ctmark) to the packet mark (nfmak) using the given masks. The new ctmark value is determined as follows:  $nfmak = (nfmak \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmak defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmak default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmak <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmak

Bits that should be cleared.

```
nfmak <0x0-0xffffffff>
```

**ctmask**

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

**log**

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

**level**

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

**prefix**

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

**tcpmss**

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

### clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

### counters (state only)

The counters of this rule.

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle prerouting rule <uint64> counters_
↳packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle prerouting rule <uint64> counters_
↳bytes
```

### input

Altering packets before routing.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle input
```

## policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle input
vrouter running input# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### Default value

accept

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle input packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle input bytes
```

## rule

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle input
vrouter running input# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
```

(continues on next page)

(continued from previous page)

```

... source \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
...   icmpv6-type [not] VALUE \
...   tcp-flags [not] set SET examined EXAMINED \
...   conntrack \
...     status [not] VALUE \
...     state [not] VALUE \
...   connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   limit burst <uint32> \
...     rate <uint32> UNIT \
...   dscp [not] VALUE \
...   tos [not] <0x0-0xff> mask <0x0-0xff> \
...   mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
...   shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
...   asconf-ack forward-tsn \
...     data examined EXAMINED set SET \
...     abort examined EXAMINED set SET \
...     shutdown-complete examined EXAMINED set SET \
...   inbound-interface [not] <string> \
...   action STANDARD chain <string> dscp DSCP \
...     connmark \
...       set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...       save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...       restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...     mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     tcpmss set-mss <uint32> clamp-mss-to-pmtu \
...     tos <0x0-0xff> mask <0x0-0xff>

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

**description**

A comment to describe the rule.

```
description <string>
```

**protocol**

Match the protocol.

```
protocol [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| ipv6-icmp    | ICMPv6 protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

## destination

Match on destination fields.

```
destination \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on destination address.

```
address [not] VALUE
```

## not

Invert the match.

```
not
```

## VALUE (mandatory)

The address to match.

```
VALUE
```

| VALUE         | Description  |
|---------------|--|
| val-ues       |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | An IPv6 address.   |
| <X:X::X/X>    | An IPv6 prefix: address and CIDR mask.   |

### port

Match on destination port.

port [not] VALUE

### not

Invert the match.

not

### VALUE (mandatory)

The port to match.

VALUE

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

### port-range

Match on destination port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

### group

Matches a set of addresses or networks.

```
group [not] <string>
```

### not

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**source**

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

**address**

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The address to match.

```
VALUE
```

| VALUE         | Description  |
|---------------|--|
| val-ues       |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | An IPv6 address.   |
| <X:X::X/X>    | An IPv6 prefix: address and CIDR mask.   |

### port

Match on source port.

port [not] VALUE

### not

Invert the match.

not

### VALUE (mandatory)

The port to match.

VALUE

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

### port-range

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

### group

Matches a set of addresses or networks.

```
group [not] <string>
```

### not

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**icmpv6-type**

Match the packet ICMP type.

```
icmpv6-type [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The ICMP type to match.

```
VALUE
```

| VALUE values               | Description   |
|----------------------------|---|
| echo-request               | Echo request.   |
| echo-reply                 | Echo reply.   |
| destination-unreachable    | Destination unreachable.                                    |
| address-unreachable        | Address unreachable.  |
| port-unreachable           | Port unreachable.   |
| no-route                   | No route to destination.                                    |
| reject-route               | Reject route to destination.                                |
| communication-prohibited   | Communication with destination administratively prohibited. |
| beyond-scope               | Beyond scope of source address.                             |
| packet-too-big             | Packet too big.   |
| failed-policy              | Source address failed ingress/egress policy.                |
| ttl-exceeded               | TTL exceeded.   |
| ttl-zero-during-transit    | Hop limit exceeded in transit.                              |
| ttl-zero-during-reassembly | Fragment reassembly time exceeded.                          |
| parameter-problem          | Parameter problem.  |
| bad-header                 | Erroneous header field encountered.                         |
| unknown-header-type        | Unrecognized Next Header type encountered.                  |
| unknown-option             | Unrecognized IPv6 option encountered.                       |
| router-solicitation        | Router solicitation.  |
| router-advertisement       | Router advertisement.                                       |
| neighbor-solicitation      | Neighbor solicitation.                                      |
| neighbor-advertisement     | Neighbor advertisement.                                     |
| redirect                   | Redirect message.   |

### tcp-flags

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

### not

Invert the match.

```
not
```

### set

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \  
shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \  
forward-tsn \  
    data examined EXAMINED set SET \  
    abort examined EXAMINED set SET \  
    shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

```
abort examined EXAMINED set SET
```

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

```
shutdown-complete examined EXAMINED set SET
```

## examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

## set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

## inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

## not

Invert the match.

```
not
```

## <string> (mandatory)

The interface to match.

```
<string>
```

### action

The action performed by this rule.

```
action STANDARD chain <string> dscp DSCP \
    connmark \
        set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
        save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
        restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
    mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    tcpmss set-mss <uint32> clamp-mss-to-pmtu \
    tos <0x0-0xff> mask <0x0-0xff>
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <string>
```

### dscp

Alters the value of the DSCP bits within the tos header of the IPv4 packet.

```
dscp DSCP
```

| DSCP values | Description   |
|-------------|---|
| <uint8>     | A differentiated services code point (DSCP) marking within the IP header. |
| af11        | AF11 (assured forwarding) class (10).                                     |
| af12        | AF12 (assured forwarding) class (12).                                     |
| af13        | AF13 (assured forwarding) class (14).                                     |
| af21        | AF21 (assured forwarding) class (18).                                     |
| af22        | AF22 (assured forwarding) class (20).                                     |
| af23        | AF23 (assured forwarding) class (22).                                     |
| af31        | AF31 (assured forwarding) class (26).                                     |
| af32        | AF32 (assured forwarding) class (28).                                     |
| af33        | AF33 (assured forwarding) class (30).                                     |
| af41        | AF41 (assured forwarding) class (34).                                     |
| af42        | AF42 (assured forwarding) class (36).                                     |
| af43        | AF43 (assured forwarding) class (38).                                     |
| be          | BE (best effort) class (0).   |
| cs0         | CS0 (class selector) class (0).   |
| cs1         | CS1 (class selector) class (8).   |
| cs2         | CS2 (class selector) class (16).  |
| cs3         | CS3 (class selector) class (24).  |
| cs4         | CS4 (class selector) class (32).  |
| cs5         | CS5 (class selector) class (40).  |
| cs6         | CS6 (class selector) class (48).  |
| cs7         | CS7 (class selector) class (56).  |
| ef          | EF (expedited forwarding) class (46).                                     |

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

XOR with this value.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

**save-mark**

Copy the packet mark (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmark \& nfmask)$  i.e. ctmask defines what bits to clear and nfmask what bits of the nfmark to XOR into the ctmark. ctmask and nfmask default to 0xFFFFFFFF.

```
save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

**nfmask**

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

**ctmask**

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

**restore-mark**

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $nfmark = (nfmark \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### nmask

Bits that should be cleared.

```
nmask <0x0-0xffffffff>
```

### ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

**prefix**

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

## set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

## clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

## tos

Alters the value of the tos header of the IPv4 packet.

```
tos <0x0-0xff> mask <0x0-0xff>
```

### <0x0-0xff> (mandatory)

Bits that should be XORed into the tos.

```
<0x0-0xff>
```

## mask

Zero the bits given by this mask in the tos.

```
mask <0x0-0xff>
```

**counters (state only)**

The counters of this rule.

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle input rule <uint64> counters packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle input rule <uint64> counters bytes
```

**forward**

Altering packets being routed.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle forward
```

**policy**

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle forward
vrouter running forward# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

**Default value**

accept

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle forward packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle forward bytes
```

**rule**

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle forward
vrouter running forward# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... source \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... icmpv6-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
... status [not] VALUE \
... state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
... rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
```

(continues on next page)

(continued from previous page)

```

...  asconf-ack forward-tsn \
...    data examined EXAMINED set SET \
...    abort examined EXAMINED set SET \
...    shutdown-complete examined EXAMINED set SET \
...  inbound-interface [not] <string> \
...  outbound-interface [not] <string> \
...  action STANDARD chain <string> \
...    connmark \
...      set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...      save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...      restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...  tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### description

A comment to describe the rule.

```
description <string>
```

### protocol

Match the protocol.

```
protocol [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

The protocol to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| ipv6-icmp    | ICMPv6 protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the ‘show filter protocols’ command or the show-filter-protocols rpc. |

### destination

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

### address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## icmpv6-type

Match the packet ICMP type.

```
icmpv6-type [not] VALUE
```

## not

Invert the match.

```
not
```

## VALUE (mandatory)

The ICMP type to match.

```
VALUE
```

| VALUE values               | Description   |
|----------------------------|---|
| echo-request               | Echo request.   |
| echo-reply                 | Echo reply.   |
| destination-unreachable    | Destination unreachable.                                    |
| address-unreachable        | Address unreachable.  |
| port-unreachable           | Port unreachable.   |
| no-route                   | No route to destination.                                    |
| reject-route               | Reject route to destination.                                |
| communication-prohibited   | Communication with destination administratively prohibited. |
| beyond-scope               | Beyond scope of source address.                             |
| packet-too-big             | Packet too big.   |
| failed-policy              | Source address failed ingress/egress policy.                |
| ttl-exceeded               | TTL exceeded.   |
| ttl-zero-during-transit    | Hop limit exceeded in transit.                              |
| ttl-zero-during-reassembly | Fragment reassembly time exceeded.                          |
| parameter-problem          | Parameter problem.  |
| bad-header                 | Erroneous header field encountered.                         |
| unknown-header-type        | Unrecognized Next Header type encountered.                  |
| unknown-option             | Unrecognized IPv6 option encountered.                       |
| router-solicitation        | Router solicitation.  |
| router-advertisement       | Router advertisement.                                       |
| neighbor-solicitation      | Neighbor solicitation.                                      |
| neighbor-advertisement     | Neighbor advertisement.                                     |
| redirect                   | Redirect message.   |

### tcp-flags

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

### not

Invert the match.

```
not
```

### set

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

### status

Match the connection status.

status [not] VALUE

### not

Invert the match.

not

### VALUE

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

### state

Match the packet state regarding conntrack.

state [not] VALUE

### not

Invert the match.

not

## VALUE

The packet states to match.

|       |
|-------|
| VALUE |
|-------|

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

|  |
|--|
| conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> |
|--|

## not

Invert the match.

|     |
|-----|
| not |
|-----|

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

|                  |
|------------------|
| <0x0-0xffffffff> |
|------------------|

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

## sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \
  shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \
  forward-tsn \
  data examined EXAMINED set SET \
  abort examined EXAMINED set SET \
  shutdown-complete examined EXAMINED set SET
```

### not

Invert the match.

```
not
```

### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

abort examined EXAMINED set SET

### examined

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

**outbound-interface**

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

**not**

Invert the match.

```
not
```

**<string> (mandatory)**

The interface to match.

```
<string>
```

**action**

The action performed by this rule.

```
action STANDARD chain <string> \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

**STANDARD**

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## chain

Jump to the user chain by this name.

```
chain <string>
```

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \  
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \  
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \  
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

## <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmark \& nfmask)$  i.e. ctmask defines what bits to clear and nfmask what bits of the nfmark to XOR into the ctmark. ctmask and nfmask default to 0xFFFFFFFF.

```
save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmask

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

## ctmask

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

## restore-mark

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $nfmark = (nfmark \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### **nmask**

Bits that should be cleared.

```
nmask <0x0-0xffffffff>
```

### **ctmask**

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### **log**

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### **level**

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

**prefix**

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

## set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

## clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

## counters (state only)

The counters of this rule.

## packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle forward rule <uint64> counters.  
↳packets
```

## bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle forward rule <uint64> counters bytes
```

## output

Altering locally-generated packets before routing.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle output
```

## policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle output
vrouter running output# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### Default value

accept

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle output packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle output bytes
```

## rule

A rule to perform an action on matching packets.

```

vrouters running config# vrf <vrf> firewall ipv6 mangle output
vrouters running output# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... source \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... icmpv6-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
...   status [not] VALUE \
...   state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
...   rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... outbound-interface [not] <string> \
... action STANDARD chain <string> dscp DSCP \
...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu \
...   tos <0x0-0xff> mask <0x0-0xff>

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### **description**

A comment to describe the rule.

```
description <string>
```

### **protocol**

Match the protocol.

```
protocol [not] VALUE
```

### **not**

Invert the match.

```
not
```

### **VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| ipv6-icmp    | ICMPv6 protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

## destination

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

## address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**icmpv6-type**

Match the packet ICMP type.

```
icmpv6-type [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The ICMP type to match.

```
VALUE
```

| VALUE values               | Description   |
|----------------------------|---|
| echo-request               | Echo request.   |
| echo-reply                 | Echo reply.   |
| destination-unreachable    | Destination unreachable.                                    |
| address-unreachable        | Address unreachable.  |
| port-unreachable           | Port unreachable.   |
| no-route                   | No route to destination.                                    |
| reject-route               | Reject route to destination.                                |
| communication-prohibited   | Communication with destination administratively prohibited. |
| beyond-scope               | Beyond scope of source address.                             |
| packet-too-big             | Packet too big.   |
| failed-policy              | Source address failed ingress/egress policy.                |
| ttl-exceeded               | TTL exceeded.   |
| ttl-zero-during-transit    | Hop limit exceeded in transit.                              |
| ttl-zero-during-reassembly | Fragment reassembly time exceeded.                          |
| parameter-problem          | Parameter problem.  |
| bad-header                 | Erroneous header field encountered.                         |
| unknown-header-type        | Unrecognized Next Header type encountered.                  |
| unknown-option             | Unrecognized IPv6 option encountered.                       |
| router-solicitation        | Router solicitation.  |
| router-advertisement       | Router advertisement.                                       |
| neighbor-solicitation      | Neighbor solicitation.                                      |
| neighbor-advertisement     | Neighbor advertisement.                                     |
| redirect                   | Redirect message.   |

### tcp-flags

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

### not

Invert the match.

```
not
```

### set

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

### status

Match the connection status.

status [not] VALUE

### not

Invert the match.

not

### VALUE

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

### state

Match the packet state regarding conntrack.

state [not] VALUE

### not

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

## sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \
  shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \
  forward-tsn \
  data examined EXAMINED set SET \
  abort examined EXAMINED set SET \
  shutdown-complete examined EXAMINED set SET
```

### not

Invert the match.

```
not
```

### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

abort examined EXAMINED set SET

### examined

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### outbound-interface

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

### action

The action performed by this rule.

```
action STANDARD chain <string> dscp DSCP \
    connmark \
        set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
        save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
        restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
    mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    tcpmss set-mss <uint32> clamp-mss-to-pmtu \
    tos <0x0-0xff> mask <0x0-0xff>
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <string>
```

### dscp

Alters the value of the DSCP bits within the tos header of the IPv4 packet.

```
dscp DSCP
```

| DSCP values | Description   |
|-------------|---|
| <uint8>     | A differentiated services code point (DSCP) marking within the IP header. |
| af11        | AF11 (assured forwarding) class (10).                                     |
| af12        | AF12 (assured forwarding) class (12).                                     |
| af13        | AF13 (assured forwarding) class (14).                                     |
| af21        | AF21 (assured forwarding) class (18).                                     |
| af22        | AF22 (assured forwarding) class (20).                                     |
| af23        | AF23 (assured forwarding) class (22).                                     |
| af31        | AF31 (assured forwarding) class (26).                                     |
| af32        | AF32 (assured forwarding) class (28).                                     |
| af33        | AF33 (assured forwarding) class (30).                                     |
| af41        | AF41 (assured forwarding) class (34).                                     |
| af42        | AF42 (assured forwarding) class (36).                                     |
| af43        | AF43 (assured forwarding) class (38).                                     |
| be          | BE (best effort) class (0).   |
| cs0         | CS0 (class selector) class (0).   |
| cs1         | CS1 (class selector) class (8).   |
| cs2         | CS2 (class selector) class (16).  |
| cs3         | CS3 (class selector) class (24).  |
| cs4         | CS4 (class selector) class (32).  |
| cs5         | CS5 (class selector) class (40).  |
| cs6         | CS6 (class selector) class (48).  |
| cs7         | CS7 (class selector) class (56).  |
| ef          | EF (expedited forwarding) class (46).                                     |

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

XOR with this value.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

**save-mark**

Copy the packet mark (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmark \& nfmask)$  i.e. ctmask defines what bits to clear and nfmask what bits of the nfmark to XOR into the ctmark. ctmask and nfmask default to 0xFFFFFFFF.

```
save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

**nfmask**

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

**ctmask**

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

**restore-mark**

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $nfmark = (nfmark \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### nmask

Bits that should be cleared.

```
nmask <0x0-0xffffffff>
```

### ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

**prefix**

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

## set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

## clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

## tos

Alters the value of the tos header of the IPv4 packet.

```
tos <0x0-0xff> mask <0x0-0xff>
```

### <0x0-0xff> (mandatory)

Bits that should be XORed into the tos.

```
<0x0-0xff>
```

## mask

Zero the bits given by this mask in the tos.

```
mask <0x0-0xff>
```

**counters (state only)**

The counters of this rule.

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle output rule <uint64> counters_
↳packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle output rule <uint64> counters bytes
```

**postrouting**

Altering packets as they are about to go.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle postrouting
```

**policy**

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle postrouting
vrouter running postrouting# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

**Default value**

accept

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle postrouting packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle postrouting bytes
```

**rule**

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle postrouting
vrouter running postrouting# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... source \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... icmpv6-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
... status [not] VALUE \
... state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
... rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
```

(continues on next page)

(continued from previous page)

```

...  asconf-ack forward-tsn \
...    data examined EXAMINED set SET \
...    abort examined EXAMINED set SET \
...    shutdown-complete examined EXAMINED set SET \
...  outbound-interface [not] <string> \
...  action STANDARD chain <string> \
...    connmark \
...      set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...      save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...      restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...    log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...    mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...    tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### description

A comment to describe the rule.

```
description <string>
```

### protocol

Match the protocol.

```
protocol [not] VALUE
```

### not

Invert the match.

```
not
```

**VALUE (mandatory)**

The protocol to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| ipv6-icmp    | ICMPv6 protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

**destination**

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

**address**

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE                            | Description  |
|----------------------------------|--|
| val-<br>ues<br><br><domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>                       | IPv6 address.  |
| <X:X::X/X>                       | IPv6 prefix: address and CIDR mask.  |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

|     |
|-----|
| not |
|-----|

**VALUE (mandatory)**

The address to match.

|       |
|-------|
| VALUE |
|-------|

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on source port.

|                  |
|------------------|
| port [not] VALUE |
|------------------|

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**icmpv6-type**

Match the packet ICMP type.

```
icmpv6-type [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The ICMP type to match.

```
VALUE
```

| VALUE values               | Description   |
|----------------------------|---|
| echo-request               | Echo request.   |
| echo-reply                 | Echo reply.   |
| destination-unreachable    | Destination unreachable.                                    |
| address-unreachable        | Address unreachable.  |
| port-unreachable           | Port unreachable.   |
| no-route                   | No route to destination.                                    |
| reject-route               | Reject route to destination.                                |
| communication-prohibited   | Communication with destination administratively prohibited. |
| beyond-scope               | Beyond scope of source address.                             |
| packet-too-big             | Packet too big.   |
| failed-policy              | Source address failed ingress/egress policy.                |
| ttl-exceeded               | TTL exceeded.   |
| ttl-zero-during-transit    | Hop limit exceeded in transit.                              |
| ttl-zero-during-reassembly | Fragment reassembly time exceeded.                          |
| parameter-problem          | Parameter problem.  |
| bad-header                 | Erroneous header field encountered.                         |
| unknown-header-type        | Unrecognized Next Header type encountered.                  |
| unknown-option             | Unrecognized IPv6 option encountered.                       |
| router-solicitation        | Router solicitation.  |
| router-advertisement       | Router advertisement.                                       |
| neighbor-solicitation      | Neighbor solicitation.                                      |
| neighbor-advertisement     | Neighbor advertisement.                                     |
| redirect                   | Redirect message.   |

### tcp-flags

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

### not

Invert the match.

```
not
```

### set

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

## sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \
  shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \
  forward-tsn \
  data examined EXAMINED set SET \
  abort examined EXAMINED set SET \
  shutdown-complete examined EXAMINED set SET
```

### not

Invert the match.

```
not
```

### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

```
abort examined EXAMINED set SET
```

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

```
shutdown-complete examined EXAMINED set SET
```

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### outbound-interface

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

## action

The action performed by this rule.

```

action STANDARD chain <string> \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

## STANDARD

Standard action.

|          |
|----------|
| STANDARD |
|----------|

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## chain

Jump to the user chain by this name.

|                |
|----------------|
| chain <string> |
|----------------|

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```

connmark \
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>

```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmask.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

### <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmark \& nfmask)$  i.e. ctmask defines what bits to clear and nfmask what bits of the nfmark to XOR into the ctmark. ctmask and nfmask default to 0xFFFFFFFF.

```
save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmask

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

## ctmask

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

## restore-mark

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $\text{nfmark} = (\text{nfmark} \& \sim\text{nfmask}) \wedge (\text{ctmark} \& \text{ctmask})$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmask

Bits that should be cleared.

```
nfmask <0x0-0xffffffff>
```

## ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

## log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

## level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

**prefix**

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

**tcpmss**

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

**set-mss**

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

**clamp-mss-to-pmtu**

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

**counters (state only)**

The counters of this rule.

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle postrouting rule <uint64> counters_
↳packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle postrouting rule <uint64> counters_
↳bytes
```

### chain

User chain.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle chain <string>
```

|          |                      |
|----------|----------------------|
| <string> | The user chain name. |
|----------|----------------------|

### policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle chain <string>
vrouter running chain <string># policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### Default value

accept

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle chain <string> packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle chain <string> bytes
```

**rule**

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv6 mangle chain <string>
vrouter running chain <string># rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... source \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... icmpv6-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
... status [not] VALUE \
... state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
... rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
```

(continues on next page)

(continued from previous page)

```

... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... inbound-interface [not] <string> \
... outbound-interface [not] <string> \
... rpfILTER invert true|false \
... action STANDARD chain <string> reject REJECT \
...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-Infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### description

A comment to describe the rule.

```
description <string>
```

### protocol

Match the protocol.

```
protocol [not] VALUE
```

### not

Invert the match.

```
not
```

**VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| ipv6-icmp    | ICMPv6 protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

**destination**

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

**address**

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

|     |
|-----|
| not |
|-----|

**VALUE (mandatory)**

The address to match.

|       |
|-------|
| VALUE |
|-------|

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on destination port.

|                  |
|------------------|
| port [not] VALUE |
|------------------|

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**icmpv6-type**

Match the packet ICMP type.

```
icmpv6-type [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The ICMP type to match.

```
VALUE
```

| VALUE values               | Description   |
|----------------------------|---|
| echo-request               | Echo request.   |
| echo-reply                 | Echo reply.   |
| destination-unreachable    | Destination unreachable.                                    |
| address-unreachable        | Address unreachable.  |
| port-unreachable           | Port unreachable.   |
| no-route                   | No route to destination.                                    |
| reject-route               | Reject route to destination.                                |
| communication-prohibited   | Communication with destination administratively prohibited. |
| beyond-scope               | Beyond scope of source address.                             |
| packet-too-big             | Packet too big.   |
| failed-policy              | Source address failed ingress/egress policy.                |
| ttl-exceeded               | TTL exceeded.   |
| ttl-zero-during-transit    | Hop limit exceeded in transit.                              |
| ttl-zero-during-reassembly | Fragment reassembly time exceeded.                          |
| parameter-problem          | Parameter problem.  |
| bad-header                 | Erroneous header field encountered.                         |
| unknown-header-type        | Unrecognized Next Header type encountered.                  |
| unknown-option             | Unrecognized IPv6 option encountered.                       |
| router-solicitation        | Router solicitation.  |
| router-advertisement       | Router advertisement.                                       |
| neighbor-solicitation      | Neighbor solicitation.                                      |
| neighbor-advertisement     | Neighbor advertisement.                                     |
| redirect                   | Redirect message.   |

### tcp-flags

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

### not

Invert the match.

```
not
```

### set

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \  
shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \  
forward-tsn \  
    data examined EXAMINED set SET \  
    abort examined EXAMINED set SET \  
    shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

abort examined EXAMINED set SET

### examined

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

## outbound-interface

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

## rpfilter

Performs a reverse path filter test on a packet. If a reply to the packet would be sent via the same interface that the packet arrived on, the packet will match.

```
rpfilter invert true|false
```

### invert

This will invert the sense of the match. Instead of matching packets that passed the reverse path filter test, match those that have failed it.

```
invert true|false
```

### Default value

false

### action

The action performed by this rule.

```
action STANDARD chain <string> reject REJECT \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <string>
```

### reject

Used to send back an error packet in response to the matched packet.

```
reject REJECT
```

| REJECT values          | Description   |
|------------------------|---|
| icmp6-no-route         | Reject with ICMPv6 no route.  |
| icmp6-adm-prohibited   | Reject with ICMPv6 admin prohibited.  |
| icmp6-addr-unreachable | Reject with ICMPv6 address unreachable.   |
| icmp6-port-unreachable | Reject with ICMPv6 port unreachable.  |
| tcp-reset              | Reject with TCP RST packet. Can be used on rules which only match the TCP protocol. |

### connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \  
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \  
  save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \  
  restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

### <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

### mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nfmak) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmak \& nfmask)$  i.e. ctmask defines what bits to clear and nfmak what bits of the nfmak to XOR into the ctmark. ctmask and nfmak default to 0xFFFFFFFF.

```
save-mark nfmak <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmak

Bits that should be XORed into the connection mark.

```
nfmak <0x0-0xffffffff>
```

## ctmask

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

## restore-mark

Copy the connection mark (ctmark) to the packet mark (nfmak) using the given masks. The new ctmark value is determined as follows:  $nfmak = (nfmak \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmak defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmak default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmak <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmak

Bits that should be cleared.

```
nfmak <0x0-0xffffffff>
```

### ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

### prefix

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

**tcpmss**

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

### clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

### counters (state only)

The counters of this rule.

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle chain <string> rule <uint64> ↵  
↳counters packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 mangle chain <string> rule <uint64> ↵  
↳counters bytes
```

### ipv6 raw

Mainly used to exempt packets from connection tracking.

```
vrouter running config# vrf <vrf> firewall ipv6 raw
```

## prerouting

Packets as soon as they come in.

```
vrouter running config# vrf <vrf> firewall ipv6 raw prerouting
```

## policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv6 raw prerouting
vrouter running prerouting# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## Default value

accept

## packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 raw prerouting packets
```

## bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 raw prerouting bytes
```

## rule

A rule to perform an action on matching packets.

```

vrouter running config# vrf <vrf> firewall ipv6 raw prerouting
vrouter running prerouting# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... source \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... icmpv6-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... contrack \
...   status [not] VALUE \
...   state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
...   rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... inbound-interface [not] <string> \
... rpfILTER invert true|false \
... action STANDARD chain <string> notrack \
...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-Infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### **description**

A comment to describe the rule.

```
description <string>
```

### **protocol**

Match the protocol.

```
protocol [not] VALUE
```

### **not**

Invert the match.

```
not
```

### **VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| ipv6-icmp    | ICMPv6 protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

## destination

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

## address

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The port to match.

VALUE

VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP.

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

port-range [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

VALUE

VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'.

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**icmpv6-type**

Match the packet ICMP type.

```
icmpv6-type [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The ICMP type to match.

```
VALUE
```

| VALUE values               | Description   |
|----------------------------|---|
| echo-request               | Echo request.   |
| echo-reply                 | Echo reply.   |
| destination-unreachable    | Destination unreachable.                                    |
| address-unreachable        | Address unreachable.  |
| port-unreachable           | Port unreachable.   |
| no-route                   | No route to destination.                                    |
| reject-route               | Reject route to destination.                                |
| communication-prohibited   | Communication with destination administratively prohibited. |
| beyond-scope               | Beyond scope of source address.                             |
| packet-too-big             | Packet too big.   |
| failed-policy              | Source address failed ingress/egress policy.                |
| ttl-exceeded               | TTL exceeded.   |
| ttl-zero-during-transit    | Hop limit exceeded in transit.                              |
| ttl-zero-during-reassembly | Fragment reassembly time exceeded.                          |
| parameter-problem          | Parameter problem.  |
| bad-header                 | Erroneous header field encountered.                         |
| unknown-header-type        | Unrecognized Next Header type encountered.                  |
| unknown-option             | Unrecognized IPv6 option encountered.                       |
| router-solicitation        | Router solicitation.  |
| router-advertisement       | Router advertisement.                                       |
| neighbor-solicitation      | Neighbor solicitation.                                      |
| neighbor-advertisement     | Neighbor advertisement.                                     |
| redirect                   | Redirect message.   |

### tcp-flags

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

### not

Invert the match.

```
not
```

### set

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

**status**

Match the connection status.

status [not] VALUE

**not**

Invert the match.

not

**VALUE**

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

**state**

Match the packet state regarding conntrack.

state [not] VALUE

**not**

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \  
shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \  
forward-tsn \  
    data examined EXAMINED set SET \  
    abort examined EXAMINED set SET \  
    shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

abort examined EXAMINED set SET

### examined

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

### examined

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### inbound-interface

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

## rpfilter

Performs a reverse path filter test on a packet. If a reply to the packet would be sent via the same interface that the packet arrived on, the packet will match.

```
rpfilter invert true|false
```

## invert

This will invert the sense of the match. Instead of matching packets that passed the reverse path filter test, match those that have failed it.

```
invert true|false
```

## Default value

false

## action

The action performed by this rule.

```
action STANDARD chain <string> notrack \
  connmark \
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

## STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## chain

Jump to the user chain by this name.

```
chain <string>
```

## notrack

Disables connection tracking for this packet.

```
notrack
```

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \  
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \  
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \  
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmask.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

## <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmark \& nfmask)$  i.e. ctmask defines what bits to clear and nfmask what bits of the nfmark to XOR into the ctmark. ctmask and nfmask default to 0xFFFFFFFF.

```
save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmask

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

## ctmask

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

## restore-mark

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $nfmark = (nfmark \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### **nfmask**

Bits that should be cleared.

```
nfmask <0x0-0xffffffff>
```

### **ctmask**

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### **log**

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### **level**

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

**prefix**

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

## set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

## clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

## counters (state only)

The counters of this rule.

## packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 raw prerouting rule <uint64> counters.  
↳packets
```

## bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 raw prerouting rule <uint64> counters bytes
```

## output

Locally-generated packets before routing.

```
vrouter running config# vrf <vrf> firewall ipv6 raw output
```

## policy

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv6 raw output
vrouter running output# policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

## Default value

accept

## packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 raw output packets
```

## bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 raw output bytes
```

## rule

A rule to perform an action on matching packets.

```

vrouters running config# vrf <vrf> firewall ipv6 raw output
vrouters running output# rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... source \
...   address [not] VALUE \
...   port [not] VALUE \
...   port-range [not] VALUE \
...   group [not] <string> \
... icmpv6-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
...   status [not] VALUE \
...   state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
...   rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... outbound-interface [not] <string> \
... action STANDARD chain <string> notrack \
...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

**description**

A comment to describe the rule.

```
description <string>
```

**protocol**

Match the protocol.

```
protocol [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| ipv6-icmp    | ICMPv6 protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

## destination

Match on destination fields.

```
destination \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on destination address.

```
address [not] VALUE
```

## not

Invert the match.

```
not
```

## VALUE (mandatory)

The address to match.

```
VALUE
```

| VALUE         | Description  |
|---------------|--|
| val-ues       |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | An IPv6 address.   |
| <X:X::X/X>    | An IPv6 prefix: address and CIDR mask.   |

### port

Match on destination port.

port [not] VALUE

### not

Invert the match.

not

### VALUE (mandatory)

The port to match.

VALUE

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

## port-range

Match on destination port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

### not

Invert the match.

```
not
```

### VALUE (mandatory)

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

### not

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**source**

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

**address**

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The address to match.

```
VALUE
```

| VALUE         | Description  |
|---------------|--|
| val-ues       |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | An IPv6 address.   |
| <X:X::X/X>    | An IPv6 prefix: address and CIDR mask.   |

### port

Match on source port.

port [not] VALUE

### not

Invert the match.

not

### VALUE (mandatory)

The port to match.

VALUE

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

## port-range

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

## not

Invert the match.

```
not
```

## VALUE (mandatory)

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**icmpv6-type**

Match the packet ICMP type.

```
icmpv6-type [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The ICMP type to match.

```
VALUE
```

| VALUE values               | Description   |
|----------------------------|---|
| echo-request               | Echo request.   |
| echo-reply                 | Echo reply.   |
| destination-unreachable    | Destination unreachable.                                    |
| address-unreachable        | Address unreachable.  |
| port-unreachable           | Port unreachable.   |
| no-route                   | No route to destination.                                    |
| reject-route               | Reject route to destination.                                |
| communication-prohibited   | Communication with destination administratively prohibited. |
| beyond-scope               | Beyond scope of source address.                             |
| packet-too-big             | Packet too big.   |
| failed-policy              | Source address failed ingress/egress policy.                |
| ttl-exceeded               | TTL exceeded.   |
| ttl-zero-during-transit    | Hop limit exceeded in transit.                              |
| ttl-zero-during-reassembly | Fragment reassembly time exceeded.                          |
| parameter-problem          | Parameter problem.  |
| bad-header                 | Erroneous header field encountered.                         |
| unknown-header-type        | Unrecognized Next Header type encountered.                  |
| unknown-option             | Unrecognized IPv6 option encountered.                       |
| router-solicitation        | Router solicitation.  |
| router-advertisement       | Router advertisement.                                       |
| neighbor-solicitation      | Neighbor solicitation.                                      |
| neighbor-advertisement     | Neighbor advertisement.                                     |
| redirect                   | Redirect message.   |

### tcp-flags

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

### not

Invert the match.

```
not
```

### set

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

### status

Match the connection status.

status [not] VALUE

### not

Invert the match.

not

### VALUE

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

### state

Match the packet state regarding conntrack.

state [not] VALUE

### not

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \  
shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \  
forward-tsn \  
    data examined EXAMINED set SET \  
    abort examined EXAMINED set SET \  
    shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

abort examined EXAMINED set SET

### examined

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

**examined**

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

**set**

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

**outbound-interface**

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

**not**

Invert the match.

```
not
```

**<string> (mandatory)**

The interface to match.

```
<string>
```

### action

The action performed by this rule.

```
action STANDARD chain <string> notrack \
    connmark \
        set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
        save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
        restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
    mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <string>
```

### notrack

Disables connection tracking for this packet.

```
notrack
```

## connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

## <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nmask) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nmask \& nmask)$  i.e. ctmask defines what bits to clear and nmask what bits of the nmask to XOR into the ctmark. ctmask and nmask default to 0xFFFFFFFF.

```
save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### **nfmask**

Bits that should be XORed into the connection mark.

```
nfmask <0x0-0xffffffff>
```

### **ctmask**

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

### **restore-mark**

Copy the connection mark (ctmark) to the packet mark (nfmark) using the given masks. The new ctmark value is determined as follows:  $\text{nfmark} = (\text{nfmark} \& \sim\text{nfmask}) \wedge (\text{ctmark} \& \text{ctmask})$  i.e. nfmask defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmask default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### **nfmask**

Bits that should be cleared.

```
nfmask <0x0-0xffffffff>
```

### **ctmask**

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

### prefix

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

### additional-infos

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

## mark

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

### <0x0-0xffffffff> (mandatory)

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

## mask

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

## tcpmss

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

### clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

**counters (state only)**

The counters of this rule.

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 raw output rule <uint64> counters packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 raw output rule <uint64> counters bytes
```

**chain**

User chain.

```
vrouter running config# vrf <vrf> firewall ipv6 raw chain <string>
```

|          |                      |
|----------|----------------------|
| <string> | The user chain name. |
|----------|----------------------|

**policy**

Action when no rule match.

```
vrouter running config# vrf <vrf> firewall ipv6 raw chain <string>
vrouter running chain <string># policy POLICY
```

| POLICY values | Description   |
|---------------|---|
| accept        | Let the packet through.   |
| drop          | Drop the packet.  |
| return        | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

**Default value**

accept

**packets (state only)**

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 raw chain <string> packets
```

**bytes (state only)**

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 raw chain <string> bytes
```

**rule**

A rule to perform an action on matching packets.

```
vrouter running config# vrf <vrf> firewall ipv6 raw chain <string>
vrouter running chain <string># rule <uint64> description <string> \
... protocol [not] VALUE \
... destination \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... source \
... address [not] VALUE \
... port [not] VALUE \
... port-range [not] VALUE \
... group [not] <string> \
... icmpv6-type [not] VALUE \
... tcp-flags [not] set SET examined EXAMINED \
... conntrack \
... status [not] VALUE \
... state [not] VALUE \
... connmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... limit burst <uint32> \
... rate <uint32> UNIT \
... dscp [not] VALUE \
... tos [not] <0x0-0xff> mask <0x0-0xff> \
... mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff> \
... sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack \
... shutdown shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf \
```

(continues on next page)

(continued from previous page)

```

... asconf-ack forward-tsn \
...   data examined EXAMINED set SET \
...   abort examined EXAMINED set SET \
...   shutdown-complete examined EXAMINED set SET \
... inbound-interface [not] <string> \
... outbound-interface [not] <string> \
... rpfilter invert true|false \
... action STANDARD chain <string> reject REJECT \
...   connmark \
...     set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...     save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...     restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...   log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
...   mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...   tcpmss set-mss <uint32> clamp-mss-to-pmtu

```

|          |   |
|----------|---|
| <uint64> | Priority of the rule. High number means lower priority. |
|----------|---|

### description

A comment to describe the rule.

```
description <string>
```

### protocol

Match the protocol.

```
protocol [not] VALUE
```

### not

Invert the match.

```
not
```

**VALUE (mandatory)**

The protocol to match.

```
VALUE
```

| VALUE values | Description   |
|--------------|---|
| tcp          | TCP protocol.   |
| udp          | UDP protocol.   |
| sctp         | SCTP protocol.  |
| ipv6-icmp    | ICMPv6 protocol.  |
| esp          | IPsec ESP protocol.   |
| ah           | IPsec AH protocol.  |
| gre          | GRE protocol.   |
| l2tp         | L2TP protocol.  |
| ipip         | IP-in-IP protocol.  |
| vrrp         | VRRP protocol.  |
| all          | All protocols.  |
| <uint16>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |
| <string>     | Protocol. The list can be obtained from the 'show filter protocols' command or the show-filter-protocols rpc. |

**destination**

Match on destination fields.

```
destination \
  address [not] VALUE \
  port [not] VALUE \
  port-range [not] VALUE \
  group [not] <string>
```

**address**

Match on destination address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on destination port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on destination port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

## group

Matches a set of addresses or networks.

```
group [not] <string>
```

## not

Not match-set.

```
not
```

## <string> (mandatory)

The name of the group.

```
<string>
```

## source

Match on source fields.

```
source \  
  address [not] VALUE \  
  port [not] VALUE \  
  port-range [not] VALUE \  
  group [not] <string>
```

## address

Match on source address.

```
address [not] VALUE
```

**not**

Invert the match.

not

**VALUE (mandatory)**

The address to match.

VALUE

| VALUE         | Description  |
|---------------|--|
| val-<br>ues   |  |
| <domain-name> | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <X:X::X:X>    | IPv6 address.  |
| <X:X::X/X>    | IPv6 prefix: address and CIDR mask.  |

**port**

Match on source port.

port [not] VALUE

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The port to match.

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|-------|---|

**port-range**

Match on source port range (syntax: port[,port|,port-port]).

```
port-range [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

Port range, syntax is port[,port|,port-port].

```
VALUE
```

|       |   |
|-------|---|
| VALUE | A comma-separated list of ports or ports ranges. Examples: '21,22,1024-2048'. |
|-------|---|

**group**

Matches a set of addresses or networks.

```
group [not] <string>
```

**not**

Not match-set.

```
not
```

**<string> (mandatory)**

The name of the group.

```
<string>
```

**icmpv6-type**

Match the packet ICMP type.

```
icmpv6-type [not] VALUE
```

**not**

Invert the match.

```
not
```

**VALUE (mandatory)**

The ICMP type to match.

```
VALUE
```

| VALUE values               | Description   |
|----------------------------|---|
| echo-request               | Echo request.   |
| echo-reply                 | Echo reply.   |
| destination-unreachable    | Destination unreachable.                                    |
| address-unreachable        | Address unreachable.  |
| port-unreachable           | Port unreachable.   |
| no-route                   | No route to destination.                                    |
| reject-route               | Reject route to destination.                                |
| communication-prohibited   | Communication with destination administratively prohibited. |
| beyond-scope               | Beyond scope of source address.                             |
| packet-too-big             | Packet too big.   |
| failed-policy              | Source address failed ingress/egress policy.                |
| ttl-exceeded               | TTL exceeded.   |
| ttl-zero-during-transit    | Hop limit exceeded in transit.                              |
| ttl-zero-during-reassembly | Fragment reassembly time exceeded.                          |
| parameter-problem          | Parameter problem.  |
| bad-header                 | Erroneous header field encountered.                         |
| unknown-header-type        | Unrecognized Next Header type encountered.                  |
| unknown-option             | Unrecognized IPv6 option encountered.                       |
| router-solicitation        | Router solicitation.  |
| router-advertisement       | Router advertisement.                                       |
| neighbor-solicitation      | Neighbor solicitation.                                      |
| neighbor-advertisement     | Neighbor advertisement.                                     |
| redirect                   | Redirect message.   |

### tcp-flags

Match the packet TCP flags.

```
tcp-flags [not] set SET examined EXAMINED
```

### not

Invert the match.

```
not
```

### set

Set flags.

```
set SET
```

| SET values | Description |
|------------|-------------|
| syn        | SYN flag.   |
| ack        | ACK flag.   |
| fin        | FIN flag.   |
| rst        | RST flag.   |
| urg        | URG flag.   |
| psh        | PSH flag.   |
| all        | All flags.  |
| none       | No flag.    |

### examined

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description |
|-----------------|-------------|
| syn             | SYN flag.   |
| ack             | ACK flag.   |
| fin             | FIN flag.   |
| rst             | RST flag.   |
| urg             | URG flag.   |
| psh             | PSH flag.   |
| all             | All flags.  |
| none            | No flag.    |

### contrack

Match contrack information.

```
contrack \  
    status [not] VALUE \  
    state [not] VALUE
```

### status

Match the connection status.

status [not] VALUE

### not

Invert the match.

not

### VALUE

The conntrack status to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| none         | No status.  |
| expected     | This is an expected connection (i.e. a conntrack helper set it up). |
| seen_reply   | Conntrack has seen packets in both directions.                      |
| assured      | Conntrack entry should never be early-expired.                      |
| confirmed    | Connection is confirmed: originating packet has left box.           |

### state

Match the packet state regarding conntrack.

state [not] VALUE

### not

Invert the match.

not

## VALUE

The packet states to match.

VALUE

| VALUE values | Description  |
|--------------|--|
| invalid      | Packet is associated with no known connection.   |
| new          | Packet started new connection or associated with one which has not seen packets in both directions.                                |
| established  | Packet is associated with a connection which has seen packets in both directions.  |
| related      | Packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer or an ICMP error. |
| untracked    | Packet is not tracked at all, which happens if you explicitly untrack it by using the notrack action in the raw table.             |
| snat         | A virtual state, matching if the original source address differs from the reply destination.                                       |
| dnat         | A virtual state, matching if the original destination differs from the reply source.   |

## conmark

Matches the mark field associated with a connection.

conmark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>

## not

Invert the match.

not

## <0x0-0xffffffff> (mandatory)

The mark value. Packets in connections are matched against this value.

<0x0-0xffffffff>

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

**limit**

Matches packets at a limited rate. If not set, the rate value is 3/hour and the burst value is 5.

```
limit burst <uint32> \  
    rate <uint32> UNIT
```

**burst**

Maximum initial number of packets to match. This number gets recharged by one every time the rate is not reached, up to this number.

```
burst <uint32>
```

**rate**

Matching rate, default unit is per hour.

```
rate <uint32> UNIT
```

**<uint32> (mandatory)**

The rate.

```
<uint32>
```

**UNIT**

Unit for rate.

```
UNIT
```

| UNIT values | Description |
|-------------|-------------|
| second      | Second.     |
| minute      | Minute.     |
| hour        | Hour.       |
| day         | Day.        |

**dscp**

Match the DSCP.

dscp [not] VALUE

**not**

Invert the match.

not

**VALUE (mandatory)**

The DSCP value to match.

VALUE

| VALUE values | Description   |
|--------------|---|
| <uint8>      | A differentiated services code point (DSCP) marking within the IP header. |
| af11         | AF11 (assured forwarding) class (10).                                     |
| af12         | AF12 (assured forwarding) class (12).                                     |
| af13         | AF13 (assured forwarding) class (14).                                     |
| af21         | AF21 (assured forwarding) class (18).                                     |
| af22         | AF22 (assured forwarding) class (20).                                     |
| af23         | AF23 (assured forwarding) class (22).                                     |
| af31         | AF31 (assured forwarding) class (26).                                     |
| af32         | AF32 (assured forwarding) class (28).                                     |
| af33         | AF33 (assured forwarding) class (30).                                     |
| af41         | AF41 (assured forwarding) class (34).                                     |
| af42         | AF42 (assured forwarding) class (36).                                     |
| af43         | AF43 (assured forwarding) class (38).                                     |
| be           | BE (best effort) class (0).   |
| cs0          | CS0 (class selector) class (0).   |
| cs1          | CS1 (class selector) class (8).   |
| cs2          | CS2 (class selector) class (16).  |
| cs3          | CS3 (class selector) class (24).  |
| cs4          | CS4 (class selector) class (32).  |
| cs5          | CS5 (class selector) class (40).  |
| cs6          | CS6 (class selector) class (48).  |
| cs7          | CS7 (class selector) class (56).  |
| ef           | EF (expedited forwarding) class (46).                                     |

**tos**

Match the tos.

```
tos [not] <0x0-0xff> mask <0x0-0xff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xff> (mandatory)**

The tos value. Packets in connections are matched against this value.

```
<0x0-0xff>
```

**mask**

Logically ANDed with the tos before the comparison.

```
mask <0x0-0xff>
```

**mark**

Matches the mark field associated with a packet.

```
mark [not] <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**not**

Invert the match.

```
not
```

**<0x0-0xffffffff> (mandatory)**

The mark value. Packets in connections are matched against this value.

```
<0x0-0xffffffff>
```

**mask**

Logically ANDed with the mark before the comparison.

```
mask <0x0-0xffffffff>
```

### sctp-chunk-types

This module matches Stream Control Transmission Protocol headers.

```
sctp-chunk-types [not] SCOPE init init-ack sack heartbeat heartbeat-ack shutdown \  
shutdown-ack error cookie-echo cookie-ack ecn-ecne ecn-cwr asconf asconf-ack \  
forward-tsn \  
    data examined EXAMINED set SET \  
    abort examined EXAMINED set SET \  
    shutdown-complete examined EXAMINED set SET
```

#### not

Invert the match.

```
not
```

#### SCOPE (mandatory)

Invert the match.

```
SCOPE
```

| SCOPE values | Description               |
|--------------|---------------------------|
| all          | Match all chunk types.    |
| any          | Match any chunk type.     |
| only         | Match exactly chunk type. |

#### init

INIT chunk.

```
init
```

**init-ack**

INIT ACK chunk.

```
init-ack
```

**sack**

SACK chunk.

```
sack
```

**heartbeat**

HEARTBEAT chunk.

```
heartbeat
```

**heartbeat-ack**

HEARTBEAT ACK chunk.

```
heartbeat-ack
```

**shutdown**

SHUTDOWN chunk.

```
shutdown
```

**shutdown-ack**

SHUTDOWN ACK chunk.

```
shutdown-ack
```

**error**

ERROR chunk.

```
error
```

**cookie-echo**

COOKIE ECHO chunk.

```
cookie-echo
```

**cookie-ack**

COOKIE ACK chunk.

```
cookie-ack
```

**ecn-ecne**

ECN ECNE chunk.

```
ecn-ecne
```

**ecn-cwr**

ECN CWR chunk.

```
ecn-cwr
```

**asconf**

ASCONF chunk.

```
asconf
```

**asconf-ack**

ASCONF ACK chunk.

```
asconf-ack
```

**forward-tsn**

FORWARD TSN chunk.

```
forward-tsn
```

**data**

DATA chunk.

```
data examined EXAMINED set SET
```

**examined**

Examined flags.

```
examined EXAMINED
```

| EXAMINED values | Description  |
|-----------------|--|
| I               | SACK chunk should be sent back without delay.  |
| U               | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B               | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E               | Marks the end fragment. An unfragmented chunk has this flag set.   |

**set**

Set flags.

```
set SET
```

| SET values | Description  |
|------------|--|
| I          | SACK chunk should be sent back without delay.  |
| U          | Indicates this data is an unordered chunk and the stream sequence number is invalid. If an unordered chunk is fragmented then each fragment has this flag set. |
| B          | Marks the beginning fragment. An unfragmented chunk has this flag set.   |
| E          | Marks the end fragment. An unfragmented chunk has this flag set.   |

### abort

ABORT chunk.

abort examined EXAMINED set SET

### examined

Examined flags.

examined EXAMINED

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

### set

Set flags.

set SET

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

### shutdown-complete

SHUTDOWN COMPLETE chunk.

shutdown-complete examined EXAMINED set SET

**examined**

Examined flags.

```
examined EXAMINED
```

|          |  |
|----------|--|
| EXAMINED | Means the sender sent its own Verification Tag (that receiver should check). |
|----------|--|

**set**

Set flags.

```
set SET
```

|     |  |
|-----|--|
| SET | Means the sender sent its own Verification Tag (that receiver should check). |
|-----|--|

**inbound-interface**

Name of an interface via which a packet was received. Only for input, forward and prerouting.

```
inbound-interface [not] <string>
```

**not**

Invert the match.

```
not
```

**<string> (mandatory)**

The interface to match.

```
<string>
```

## outbound-interface

Name of an interface via which a packet is going to be sent. Only for forward, output and postrouting.

```
outbound-interface [not] <string>
```

### not

Invert the match.

```
not
```

### <string> (mandatory)

The interface to match.

```
<string>
```

## rpfilter

Performs a reverse path filter test on a packet. If a reply to the packet would be sent via the same interface that the packet arrived on, the packet will match.

```
rpfilter invert true|false
```

### invert

This will invert the sense of the match. Instead of matching packets that passed the reverse path filter test, match those that have failed it.

```
invert true|false
```

### Default value

false

### action

The action performed by this rule.

```
action STANDARD chain <string> reject REJECT \
  connmark \
    set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
    save-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
    restore-mark nmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
  log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS \
  mark <0x0-0xffffffff> mask <0x0-0xffffffff> \
  tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### STANDARD

Standard action.

```
STANDARD
```

| STANDARD values | Description   |
|-----------------|---|
| accept          | Let the packet through.   |
| drop            | Drop the packet.  |
| return          | Stop traversing this chain and resume at the next rule in the parent chain. For built-ins, go through the policy. |

### chain

Jump to the user chain by this name.

```
chain <string>
```

### reject

Used to send back an error packet in response to the matched packet.

```
reject REJECT
```

| REJECT values          | Description   |
|------------------------|---|
| icmp6-no-route         | Reject with ICMPv6 no route.  |
| icmp6-adm-prohibited   | Reject with ICMPv6 admin prohibited.  |
| icmp6-addr-unreachable | Reject with ICMPv6 address unreachable.   |
| icmp6-port-unreachable | Reject with ICMPv6 port unreachable.  |
| tcp-reset              | Reject with TCP RST packet. Can be used on rules which only match the TCP protocol. |

### connmark

Sets the mark value associated with a connection. The mark is 32 bits wide.

```
connmark \  
  set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \  
  save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \  
  restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

### set-xmark

Zero out the bits given by mask and XOR value into the ctmark.

```
set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

### <0x0-0xffffffff> (mandatory)

XOR with this value.

```
<0x0-0xffffffff>
```

### mask

Zero the bits given by this mask.

```
mask <0x0-0xffffffff>
```

## save-mark

Copy the packet mark (nfmak) to the connection mark (ctmark) using the given masks. The new value is determined as follows:  $ctmark = (ctmark \& \sim ctmask) \wedge (nfmak \& nfmask)$  i.e. ctmask defines what bits to clear and nfmak what bits of the nfmak to XOR into the ctmark. ctmask and nfmak default to 0xFFFFFFFF.

```
save-mark nfmak <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmak

Bits that should be XORed into the connection mark.

```
nfmak <0x0-0xffffffff>
```

## ctmask

Bits that should be cleared.

```
ctmask <0x0-0xffffffff>
```

## restore-mark

Copy the connection mark (ctmark) to the packet mark (nfmak) using the given masks. The new ctmark value is determined as follows:  $nfmak = (nfmak \& \sim nfmask) \wedge (ctmark \& ctmask)$  i.e. nfmak defines what bits to clear and ctmask what bits of the ctmark to XOR into the packet mark. ctmask and nfmak default to 0xFFFFFFFF. restore-mark is only valid in the mangle table.

```
restore-mark nfmak <0x0-0xffffffff> ctmask <0x0-0xffffffff>
```

## nfmak

Bits that should be cleared.

```
nfmak <0x0-0xffffffff>
```

### ctmask

Bits that should be XORed into the packet mark.

```
ctmask <0x0-0xffffffff>
```

### log

Turn on logging of matching packets.

```
log level LEVEL prefix <string> additional-infos ADDITIONAL-INFOS
```

### level

Level of logging.

```
level LEVEL
```

| LEVEL values | Description      |
|--------------|------------------|
| emergency    | Emergency level. |
| alert        | Alert level.     |
| critical     | Critical level.  |
| error        | Error level.     |
| warning      | Warning level.   |
| notice       | Notice level.    |
| info         | Info level.      |
| debug        | Debug level.     |

### prefix

Prefix log messages with the specified prefix, up to 29 letters long.

```
prefix <string>
```

**additional-infos**

Append additional informations to the logs.

```
additional-infos ADDITIONAL-INFOS
```

| ADDITIONAL-INFOS values | Description   |
|-------------------------|---|
| tcp-sequence            | Log TCP sequence numbers.                                 |
| tcp-options             | Log options from the TCP packet header.                   |
| ip-options              | Log options from the IP/IPv6 packet header.               |
| user-id                 | Log the userid of the process which generated the packet. |

**mark**

Used to set the mark value associated with the packet.

```
mark <0x0-0xffffffff> mask <0x0-0xffffffff>
```

**<0x0-0xffffffff> (mandatory)**

Bits that should be XORed into the packet mark.

```
<0x0-0xffffffff>
```

**mask**

Zero the bits given by this mask in the packet mark.

```
mask <0x0-0xffffffff>
```

**tcpmss**

Alters the MSS value of TCP SYN packets, to control the maximum size for that connection.

```
tcpmss set-mss <uint32> clamp-mss-to-pmtu
```

### set-mss

Explicitly sets MSS option to specified value.

```
set-mss <uint32>
```

### clamp-mss-to-pmtu

Automatically clamp MSS value to (path\_MTU - 40 for IPv4, - 60 for IPv6).

```
clamp-mss-to-pmtu
```

### counters (state only)

The counters of this rule.

### packets (state only)

Packets.

```
vrouter> show state vrf <vrf> firewall ipv6 raw chain <string> rule <uint64> counters_
↳packets
```

### bytes (state only)

Bytes.

```
vrouter> show state vrf <vrf> firewall ipv6 raw chain <string> rule <uint64> counters_
↳bytes
```

### ipv6 address group

**Attention:**

**Deprecated since:** 2021-01-04

**Obsolete in release:** 21q3

**Description:** This configuration has been moved outside of the firewall configuration to be usable in other contexts.

**Replacement:** / vrf group ipv6 address-group

Address group.

```
vrouter running config# vrf <vrf> firewall ipv6 address-group <string>
```

|          |                            |
|----------|----------------------------|
| <string> | Name of the address group. |
|----------|----------------------------|

### address (deprecated)

List of addresses of the group.

```
vrouter running config# vrf <vrf> firewall ipv6 address-group <string>
vrouter running address-group <string># address ADDRESS
```

|              |   |
|--------------|---|
| AD-<br>DRESS | An IPv6 address without a zone index. This type, derived from ipv6-address, may be used in situations where the zone is known from the context and hence no zone index is needed. |
|--------------|---|

### ipv6 network group

**Attention:**

**Deprecated since:** 2021-01-04

**Obsolete in release:** 21q3

**Description:** This configuration has been moved outside of the firewall configuration to be usable in other contexts.

**Replacement:** / vrf group ipv6 network-group

Network group.

```
vrouter running config# vrf <vrf> firewall ipv6 network-group <string>
```

|          |                            |
|----------|----------------------------|
| <string> | Name of the network group. |
|----------|----------------------------|

### network (deprecated)

List of networks of the group.

```
vrouter running config# vrf <vrf> firewall ipv6 network-group <string>
vrouter running network-group <string># network NETWORK
```

|         |  |
|---------|--|
| NETWORK | An IPv6 prefix: address and CIDR mask. |
|---------|--|

### 3.2.21 network-port (state only)

The list of network ports on the device.

#### pci-bus-addr (state only)

The bus address of the PCI device.

```
vrouter> show state network-port <string> pci-bus-addr
```

#### device-tree-alias (state only)

In case of device tree port, the alias if there is one.

```
vrouter> show state network-port <string> device-tree-alias
```

#### device-tree-compatible (state only)

In case of device tree port, the compatible field if there is one.

```
vrouter> show state network-port <string> device-tree-compatible
```

#### vendor (state only)

The device vendor.

```
vrouter> show state network-port <string> vendor
```

### model (state only)

The device model.

```
vrouter> show state network-port <string> model
```

### device-port (state only)

The port number, in case there are several ports per PCI device.

```
vrouter> show state network-port <string> device-port
```

### mac-address (state only)

The port MAC address.

```
vrouter> show state network-port <string> mac-address
```

### interface (state only)

The interface name.

```
vrouter> show state network-port <string> interface
```

## 3.2.22 interface

### bridge

The list of bridge interfaces on the device.

```
vrouter running config# vrf <vrf> interface bridge <bridge>
```

|          |                    |
|----------|--------------------|
| <bridge> | An interface name. |
|----------|--------------------|

## mtu

Set the max transmission unit size in octets.

```
vrouter running config# vrf <vrf> interface bridge <bridge>  
vrouter running bridge <bridge># mtu <uint32>
```

## promiscuous

Set promiscuous mode.

```
vrouter running config# vrf <vrf> interface bridge <bridge>  
vrouter running bridge <bridge># promiscuous true|false
```

## description

A textual description of the interface.

```
vrouter running config# vrf <vrf> interface bridge <bridge>  
vrouter running bridge <bridge># description <string>
```

## enabled

The desired (administrative) state of the interface.

```
vrouter running config# vrf <vrf> interface bridge <bridge>  
vrouter running bridge <bridge># enabled true|false
```

## Default value

true

## ageing-time

Configure the bridge's FDB entries ageing time, ie the time a MAC address will be kept in the FDB after a packet has been received from that address. After this time has passed, entries are cleaned up.

```
vrouter running config# vrf <vrf> interface bridge <bridge>  
vrouter running bridge <bridge># ageing-time <uint32>
```

### ifindex (state only)

System assigned number for each interface. Corresponds to ifIndex object in SNMP Interface MIB.

```
vrouter> show state vrf <vrf> interface bridge <bridge> ifindex
```

### admin-status (state only)

The desired state of the interface. In RFC 7223 this leaf has the same read semantics as ifAdminStatus. Here, it reflects the administrative state as set by enabling or disabling the interface.

```
vrouter> show state vrf <vrf> interface bridge <bridge> admin-status
```

### oper-status (state only)

The current operational state of the interface. This leaf has the same semantics as ifOperStatus.

```
vrouter> show state vrf <vrf> interface bridge <bridge> oper-status
```

### last-change (state only)

This timestamp indicates the time of the last state change of the interface (e.g., up-to-down transition). This corresponds to the ifLastChange object in the standard interface MIB. The value is the timestamp in nanoseconds relative to the Unix Epoch (Jan 1, 1970 00:00:00 UTC).

```
vrouter> show state vrf <vrf> interface bridge <bridge> last-change
```

### ethernet

Top-level container for Ethernet configuration.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ethernet
```

## mac-address

Assigns a MAC address to the Ethernet interface. If not specified, the corresponding operational state leaf is expected to show the system-assigned MAC address.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ethernet
vrouter running ethernet# mac-address MAC-ADDRESS
```

|             |  |
|-------------|--|
| MAC-ADDRESS | An IEEE 802 unicast MAC address i.e. the second digit is an even number. Moreover the mac address must not be 00:00:00:00:00:00. |
|-------------|--|

## ipv4

Parameters for the IPv4 address family.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4
```

## enabled

Controls whether IPv4 is enabled or disabled on this interface. When IPv4 is enabled, this interface is connected to an IPv4 stack, and the interface can send and receive IPv4 packets.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4
vrouter running ipv4# enabled true|false
```

## Default value

true

## address

The list of configured IPv4 addresses on the interface.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4
vrouter running ipv4# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <A.B.C.D>        | An IPv4 address.  |
| <A.B.C.D/M>      | A masked IPv4 address: address and prefix of that subnet. |

**peer**

The IPv4 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv4 address. |
|------|------------------|

**origin (state only)**

The origin of this address, e.g., statically configured, assigned by DHCP, etc..

```
vrouter> show state vrf <vrf> interface bridge <bridge> ipv4 address <address> origin
```

**neighbor**

A list of mappings from IPv4 addresses to link-layer addresses. Entries in this list are used as static entries in the ARP Cache.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4
vrouter running ipv4# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv4 address. |
|------------|------------------|

**link-layer-address (mandatory)**

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

## state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface bridge <bridge> ipv4 neighbor <neighbor> state
```

## dhcp

DHCP client configuration.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4 dhcp
```

## enabled

Enable or disable DHCP.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4 dhcp  
vrouter running dhcp# enabled true|false
```

## Default value

true

## timeout

Time before deciding that it's not going to be able to contact a server.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4 dhcp  
vrouter running dhcp# timeout <uint32>
```

## Default value

60

## retry

Time before trying again to contact a DHCP server.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4 dhcp  
vrouter running dhcp# retry <uint32>
```

## Default value

300

### select-timeout

Time at which the client stops waiting for other offers from servers.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4 dhcp  
vrouter running dhcp# select-timeout <uint32>
```

#### Default value

0

### reboot

Time after trying to reacquire its old address before trying to discover a new address.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4 dhcp  
vrouter running dhcp# reboot <uint32>
```

#### Default value

10

### initial-interval

Time between the first attempt to reach a server and the second attempt to reach a server.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4 dhcp  
vrouter running dhcp# initial-interval <uint32>
```

#### Default value

10

### dhcp-lease-time

Requested lease time.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4 dhcp  
vrouter running dhcp# dhcp-lease-time <uint32>
```

#### Default value

7200

### dhcp-client-identifier-ascii

DHCP client identifier (ASCII).

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4 dhcp  
vrouter running dhcp# dhcp-client-identifier-ascii <string>
```

### dhcp-client-identifier-hexa

DHCP client identifier (hexadecimal).

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4 dhcp  
vrouter running dhcp# dhcp-client-identifier-hexa <string>
```

### host-name

DHCP client name.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4 dhcp  
vrouter running dhcp# host-name <string>
```

### request

DHCP requests.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv4 dhcp  
vrouter running dhcp# request REQUEST
```

| REQUEST values       | Description  |
|----------------------|--|
| subnet-mask          | Client's subnet mask.  |
| broadcast-address    | Broadcast address in use on the client's subnet.                     |
| time-offset          | Offset of the client's subnet in seconds from UTC.                   |
| routers              | List of IP addresses for routers on the client's subnet.             |
| domain-name          | Domain name used when resolving hostnames with DNS.                  |
| domain-search        | Domain search list used when resolving hostnames with DNS.           |
| domain-name-servers  | List of DNS name servers available to the client.                    |
| host-name            | Name of the client.  |
| nis-domain           | Name of the client's NIS (Sun Network Information Services) domain.  |
| nis-servers          | List of IP addresses indicating NIS servers available to the client. |
| ntp-servers          | List of IP addresses indicating NTP servers available to the client. |
| interface-mtu        | MTU to use on this interface.  |
| netbios-name-servers | List of RFC 1001/1002 NBNS name servers.                             |
| netbios-scope        | NetBIOS over TCP/IP scope parameter for the client.                  |

### Default value

subnet-mask  
 broadcast-address  
 time-offset  
 routers  
 domain-name  
 domain-search  
 domain-name-servers  
 host-name  
 nis-domain  
 nis-servers  
 ntp-servers  
 interface-mtu

### current-lease (state only)

Current lease.

### fixed-address (state only)

The IPv4 address on the interface.

```
vrouter> show state vrf <vrf> interface bridge <bridge> ipv4 dhcp current-lease fixed-  
↪address
```

### renew (state only)

Time at which the client should begin trying to contact its server to renew its lease.

```
vrouter> show state vrf <vrf> interface bridge <bridge> ipv4 dhcp current-lease renew
```

### rebind (state only)

Time at which the client should begin to try to contact any dhcp server to renew its lease.

```
vrouter> show state vrf <vrf> interface bridge <bridge> ipv4 dhcp current-lease rebind
```

### expire (state only)

Time at which the client must stop using a lease if it has not been able to renew it.

```
vrouter> show state vrf <vrf> interface bridge <bridge> ipv4 dhcp current-lease expire
```

## ipv6

Parameters for the IPv6 address family.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv6
```

### enabled

Controls whether IPv6 is enabled or disabled on this interface. When IPv6 is enabled, this interface is connected to an IPv6 stack, and the interface can send and receive IPv6 packets.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv6  
vrouter running ipv6# enabled true|false
```

### Default value

true

### address

The list of configured IPv6 addresses on the interface.

```
vrouter running config# vrf <vrf> interface bridge <bridge> ipv6
vrouter running ipv6# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <X:X::X:X>       | An IPv6 address.  |
| <X:X::X:X/M>     | A masked IPv6 address: address and prefix of that subnet. |

### peer

The IPv6 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv6 address. |
|------|------------------|

### origin (state only)

The origin of this address, e.g., static, dhcp, etc.

```
vrouter> show state vrf <vrf> interface bridge <bridge> ipv6 address <address> origin
```

### status (state only)

The status of an address. Most of the states correspond to states from the IPv6 Stateless Address Autoconfiguration protocol.

```
vrouter> show state vrf <vrf> interface bridge <bridge> ipv6 address <address> status
```

## neighbor

List of IPv6 neighbors.

```

vrouters running config# vrf <vrf> interface bridge <bridge> ipv6
vrouters running ipv6# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS

```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv6 address. |
|------------|------------------|

### link-layer-address (mandatory)

The link-layer address of the neighbor node.

```

link-layer-address LINK-LAYER-ADDRESS

```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

### router (state only)

Indicates that the neighbor node acts as a router.

```

vrouters> show state vrf <vrf> interface bridge <bridge> ipv6 neighbor <neighbor> router

```

### state (state only)

The state of this neighbor entry.

```

vrouters> show state vrf <vrf> interface bridge <bridge> ipv6 neighbor <neighbor> state

```

### network-stack

Network stack parameters for this interface.

```

vrouters running config# vrf <vrf> interface bridge <bridge> network-stack

```

## ipv4

IPv4 parameters.

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv4
```

### send-redirects

Send ICMP redirect if host is on the same network than gateway.

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv4  
vrouter running ipv4# send-redirects true|false
```

### accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv4  
vrouter running ipv4# accept-redirects true|false
```

### accept-source-route

Accept packets with source route option. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv4  
vrouter running ipv4# accept-source-route true|false
```

### arp-announce

Define different restriction levels for announcing the local source IP address from IP packets in ARP requests sent on interface. Increasing the restriction level gives more chance for receiving answer from the resolved target while decreasing the level announces more valid sender's information.

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv4  
vrouter running ipv4# arp-announce ARP-ANNOUNCE
```

| ARP-AND-Description<br>val-ues | Description  |
|--------------------------------|--|
| any                            | Use any local address, configured on any interface.  |
| avoid-not-in-subnet            | Try to avoid local addresses that are not in the target's subnet for this interface. This mode is useful when target hosts reachable via this interface require the source IP address in ARP requests to be part of their logical network configured on the receiving interface. When we generate the request we will check all our subnets that include the target IP and will preserve the source address if it is from such subnet. If there is no such subnet we select source address according to the rules for level 2, 'best-local'.   |
| best-local                     | Always use the best local address for this target. In this mode we ignore the source address in the IP packet and try to select local address that we prefer for talks with the target host. Such local address is selected by looking for primary IP addresses on all our subnets on the outgoing interface that include the target IP address. If no suitable local address is found we select the first local address we have on the outgoing interface or on all other interfaces, with the hope we will receive reply for our request and even sometimes no matter the source IP address we announce. |

### arp-filter

Allows to have multiple network interfaces on the same subnet, and have the ARPs for each interface be answered based on whether or not the kernel would route a packet from the ARP'd IP out that interface (therefore you must use source based routing for this to work). In other words it allows control of which cards (usually 1) will respond to an arp request.

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv4
vrouter running ipv4# arp-filter true|false
```

### arp-ignore

Define different modes for sending replies in response to received ARP requests that resolve local target IP addresses.

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv4
vrouter running ipv4# arp-ignore ARP-IGNORE
```

| ARP-IGNORE values          | Description  |
|----------------------------|--|
| any                        | Reply for any local target IP address, configured on any interface.  |
| check-interface            | Reply only if the target IP address is local address configured on the incoming interface.   |
| check-interface-and-subnet | Reply only if the target IP address is local address configured on the incoming interface and both with the sender's IP address are part from same subnet on this interface. |
| ignore-scope               | Do not reply for local addresses configured with scope host, only resolutions for global and link addresses are replied.   |
| ignore-all                 | Do not reply for all local addresses.  |

### log-invalid-addresses

Log packets with impossible addresses.

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv4
vrouter running ipv4# log-invalid-addresses true|false
```

### ipv6

IPv6 parameters.

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv6
```

### autoconfiguration

Autoconfigure addresses using Prefix Information in Router Advertisements.

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv6
vrouter running ipv6# autoconfiguration true|false
```

### accept-router-advert

Accept Router Advertisements.

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv6
vrouter running ipv6# accept-router-advert ACCEPT-ROUTER-ADVERT
```

| ACCEPT-ROUTER-ADVERT values | Description   |
|-----------------------------|---|
| never                       | Do not accept Router Advertisements.                        |
| norouter-mode               | Accept Router Advertisements if forwarding is disabled.     |
| always                      | Accept Router Advertisements even if forwarding is enabled. |

### accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router.

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv6
vrouter running ipv6# accept-redirects true|false
```

### accept-source-route

Accept packets with source route option.

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv6
vrouter running ipv6# accept-source-route true|false
```

### router-solicitations

Number of Router Solicitations to send until assuming no routers are present.

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv6
vrouter running ipv6# router-solicitations <int16>
```

### use-temporary-addresses

Preference for Privacy Extensions (RFC4941). Not applied to point-to-point and loopback devices (always 0).

```
vrouter running config# vrf <vrf> interface bridge <bridge> network-stack ipv6
vrouter running ipv6# use-temporary-addresses USE-TEMPORARY-ADDRESSES
```

| USE-TEMPORARY-ADDRESS values | Description   |
|------------------------------|---|
| never                        | Disable Privacy Extensions, i.e. use the public address, subnet prefix/interface id, where interface id is always the same. |
| prefer-public-addresses      | Enable Privacy Extensions, but prefer public addresses over temporary addresses.  |
| always                       | Enable Privacy Extensions and prefer temporary addresses over public addresses.   |

## stp

Configure spanning tree protocol parameters.

```
vrouter running config# vrf <vrf> interface bridge <bridge> stp
```

## enabled

Enable or disable spanning tree protocol on this bridge.

```
vrouter running config# vrf <vrf> interface bridge <bridge> stp  
vrouter running stp# enabled true|false
```

## Default value

true

## priority

Set this bridge's spanning tree priority, used during STP root bridge election.

```
vrouter running config# vrf <vrf> interface bridge <bridge> stp  
vrouter running stp# priority <uint16>
```

## forward-delay

Set the forwarding delay, ie the time spent in LISTENING state (before moving to LEARNING) and in LEARNING state (before moving to FORWARDING).

```
vrouter running config# vrf <vrf> interface bridge <bridge> stp  
vrouter running stp# forward-delay <uint32>
```

## max-age

Set the hello packet timeout, ie the time until another bridge in the spanning tree is assumed to be dead, after reception of its last hello message.

```
vrouter running config# vrf <vrf> interface bridge <bridge> stp  
vrouter running stp# max-age <uint32>
```

## hello-time

Set the time between hello packets sent by the bridge, when it is a root bridge or a designated bridges.

```
vrouter running config# vrf <vrf> interface bridge <bridge> stp  
vrouter running stp# hello-time <uint32>
```

## link-interface

Set this interface as slave of this bridge.

```
vrouter running config# vrf <vrf> interface bridge <bridge>  
vrouter running bridge <bridge># link-interface <link-interface> learning true|false
```

|                  |                    |
|------------------|--------------------|
| <link-interface> | An interface name. |
|------------------|--------------------|

## learning

Allow MAC address learning on this slave.

```
learning true|false
```

### Default value

true

## qos

---

**Note:** requires a Turbo Router Network License.

---

QoS configuration.

```
vrouter running config# vrf <vrf> interface bridge <bridge> qos
```

## ingress

Ingress QoS configuration.

```
vrouter running config# vrf <vrf> interface bridge <bridge> qos ingress
```

## rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface bridge <bridge> qos ingress rate-limit
```

## policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface bridge <bridge> qos ingress rate-limit  
vrouter running rate-limit# policer <leafref>
```

## shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface bridge <bridge> qos ingress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

## policer (state only)

Traffic policer.

## bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos ingress rate-limit policer ↵  
↵ bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos ingress rate-limit policer.  
↳burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos ingress rate-limit policer.  
↳excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos ingress rate-limit policer.  
↳excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos ingress rate-limit policer.  
↳shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos ingress rate-limit policer_↵  
↵stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos ingress rate-limit policer_↵  
↵stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos ingress rate-limit policer_↵  
↵stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos ingress rate-limit policer_↵  
↵stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos ingress rate-limit policer_↵  
↵stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos ingress rate-limit policer.  
↳stats drop-bytes
```

### egress

Egress QoS configuration.

```
vrouter running config# vrf <vrf> interface bridge <bridge> qos egress
```

### rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface bridge <bridge> qos egress rate-limit
```

### policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface bridge <bridge> qos egress rate-limit  
vrouter running rate-limit# policer <leafref>
```

### shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface bridge <bridge> qos egress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

### **policer (state only)**

Traffic policer.

### **bandwidth (state only)**

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos egress rate-limit policer_
↳bandwidth
```

### **burst (state only)**

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos egress rate-limit policer_
↳burst
```

### **excess-bandwidth (state only)**

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos egress rate-limit policer_
↳excess-bandwidth
```

### **excess-burst (state only)**

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos egress rate-limit policer_
↳excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos egress rate-limit policer_↵  
↵shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos egress rate-limit policer_↵  
↵stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos egress rate-limit policer_↵  
↵stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos egress rate-limit policer_↵  
↵stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos egress rate-limit policer_↵  
↵stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos egress rate-limit policer.  
↳stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface bridge <bridge> qos egress rate-limit policer.  
↳stats drop-bytes
```

### counters (state only)

A collection of interface-related statistics objects.

### in-octets (state only)

The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface bridge <bridge> counters in-octets
```

### in-unicast-pkts (state only)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface bridge <bridge> counters in-unicast-pkts
```

### in-discards (state only)

The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface bridge <bridge> counters in-discards
```

### in-errors (state only)

For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface bridge <bridge> counters in-errors
```

### out-octets (state only)

The total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface bridge <bridge> counters out-octets
```

### out-unicast-pkts (state only)

The total number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface bridge <bridge> counters out-unicast-pkts
```

### out-discards (state only)

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface bridge <bridge> counters out-discards
```

### out-errors (state only)

For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface bridge <bridge> counters out-errors
```

### gre

---

**Note:** requires a Turbo Router Network License.

---

The list of GRE interfaces on the device.

```
vrouter running config# vrf <vrf> interface gre <gre>
```

|       |                    |
|-------|--------------------|
| <gre> | An interface name. |
|-------|--------------------|

### mtu

Set the max transmission unit size in octets.

```
vrouter running config# vrf <vrf> interface gre <gre>  
vrouter running gre <gre># mtu <uint32>
```

## promiscuous

Set promiscuous mode.

```
vrouter running config# vrf <vrf> interface gre <gre>  
vrouter running gre <gre># promiscuous true|false
```

## description

A textual description of the interface.

```
vrouter running config# vrf <vrf> interface gre <gre>  
vrouter running gre <gre># description <string>
```

## enabled

The desired (administrative) state of the interface.

```
vrouter running config# vrf <vrf> interface gre <gre>  
vrouter running gre <gre># enabled true|false
```

## Default value

true

## ttl

The time-to-live (or hop limit) that should be utilised for the IP packets used for the tunnel transport.

```
vrouter running config# vrf <vrf> interface gre <gre>  
vrouter running gre <gre># ttl <uint8>
```

## tos

Set the DSCP bits in the Type of Service field.

```
vrouter running config# vrf <vrf> interface gre <gre>  
vrouter running gre <gre># tos <uint8>
```

## link-interface

Route tunneled packets through this interface.

```
vrouter running config# vrf <vrf> interface gre <gre>
vrouter running gre <gre># link-interface LINK-INTERFACE
```

|                |                    |
|----------------|--------------------|
| LINK-INTERFACE | An interface name. |
|----------------|--------------------|

## link-vrf

The link vrf name.

```
vrouter running config# vrf <vrf> interface gre <gre>
vrouter running gre <gre># link-vrf <string>
```

## local (mandatory)

The source address that should be used for the tunnel.

```
vrouter running config# vrf <vrf> interface gre <gre>
vrouter running gre <gre># local LOCAL
```

| LOCAL values | Description      |
|--------------|------------------|
| <A.B.C.D>    | An IPv4 address. |
| <X:X::X:X>   | An IPv6 address. |

## remote

The destination address that should be used for the tunnel.

```
vrouter running config# vrf <vrf> interface gre <gre>
vrouter running gre <gre># remote REMOTE
```

| REMOTE values | Description      |
|---------------|------------------|
| <A.B.C.D>     | An IPv4 address. |
| <X:X::X:X>    | An IPv6 address. |

## checksum

Enable checksum features for this tunnel.

```
vrouter running config# vrf <vrf> interface gre <gre>
vrouter running gre <gre># checksum CHECKSUM
```

| CHECKSUM values | Description   |
|-----------------|---|
| input           | Verify checksum for all input packets.  |
| output          | Calculate checksum for outgoing packets.                                      |
| both            | Calculate checksum for outgoing packets, and verify it for all input packets. |

## sequence-number

Enable sequence number for this tunnel.

```
vrouter running config# vrf <vrf> interface gre <gre>
vrouter running gre <gre># sequence-number SEQUENCE-NUMBER
```

| SEQUENCE-NUMBER values | Description  |
|------------------------|--|
| input                  | All input packet must be serialized.   |
| output                 | Enable sequencing of outgoing packets.   |
| both                   | Enable sequencing of outgoing packet and check serialization of all input packets. |

## ifindex (state only)

System assigned number for each interface. Corresponds to ifIndex object in SNMP Interface MIB.

```
vrouter> show state vrf <vrf> interface gre <gre> ifindex
```

## admin-status (state only)

The desired state of the interface. In RFC 7223 this leaf has the same read semantics as ifAdminStatus. Here, it reflects the administrative state as set by enabling or disabling the interface.

```
vrouter> show state vrf <vrf> interface gre <gre> admin-status
```

### oper-status (state only)

The current operational state of the interface. This leaf has the same semantics as ifOperStatus.

```
vrouter> show state vrf <vrf> interface gre <gre> oper-status
```

### last-change (state only)

This timestamp indicates the time of the last state change of the interface (e.g., up-to-down transition). This corresponds to the ifLastChange object in the standard interface MIB. The value is the timestamp in nanoseconds relative to the Unix Epoch (Jan 1, 1970 00:00:00 UTC).

```
vrouter> show state vrf <vrf> interface gre <gre> last-change
```

### ipv4

Parameters for the IPv4 address family.

```
vrouter running config# vrf <vrf> interface gre <gre> ipv4
```

### enabled

Controls whether IPv4 is enabled or disabled on this interface. When IPv4 is enabled, this interface is connected to an IPv4 stack, and the interface can send and receive IPv4 packets.

```
vrouter running config# vrf <vrf> interface gre <gre> ipv4
vrouter running ipv4# enabled true|false
```

### Default value

true

### address

The list of configured IPv4 addresses on the interface.

```
vrouter running config# vrf <vrf> interface gre <gre> ipv4
vrouter running ipv4# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <A.B.C.D>        | An IPv4 address.  |
| <A.B.C.D/M>      | A masked IPv4 address: address and prefix of that subnet. |

**peer**

The IPv4 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv4 address. |
|------|------------------|

**origin (state only)**

The origin of this address, e.g., statically configured, assigned by DHCP, etc..

```
vrouter> show state vrf <vrf> interface gre <gre> ipv4 address <address> origin
```

**neighbor**

A list of mappings from IPv4 addresses to link-layer addresses.

```
vrouter running config# vrf <vrf> interface gre <gre> ipv4
vrouter running ipv4# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv4 address. |
|------------|------------------|

**link-layer-address (mandatory)**

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

| LINK-LAYER-ADDRESS values | Description      |
|---------------------------|------------------|
| <A.B.C.D>                 | An IPv4 address. |
| <X:X::X:X>                | An IPv6 address. |

**state (state only)**

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface gre <gre> ipv4 neighbor <neighbor> state
```

**ipv6**

Parameters for the IPv6 address family.

```
vrouter running config# vrf <vrf> interface gre <gre> ipv6
```

**enabled**

Controls whether IPv6 is enabled or disabled on this interface. When IPv6 is enabled, this interface is connected to an IPv6 stack, and the interface can send and receive IPv6 packets.

```
vrouter running config# vrf <vrf> interface gre <gre> ipv6
vrouter running ipv6# enabled true|false
```

**Default value**

true

**address**

The list of configured IPv6 addresses on the interface.

```
vrouter running config# vrf <vrf> interface gre <gre> ipv6
vrouter running ipv6# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <X:X::X:X>       | An IPv6 address.  |
| <X:X::X:X/M>     | A masked IPv6 address: address and prefix of that subnet. |

**peer**

The IPv6 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv6 address. |
|------|------------------|

**origin (state only)**

The origin of this address, e.g., static, dhcp, etc.

```
vrouter> show state vrf <vrf> interface gre <gre> ipv6 address <address> origin
```

**status (state only)**

The status of an address. Most of the states correspond to states from the IPv6 Stateless Address Autoconfiguration protocol.

```
vrouter> show state vrf <vrf> interface gre <gre> ipv6 address <address> status
```

**neighbor**

A list of mappings from IPv6 addresses to link-layer addresses.

```
vrouter running config# vrf <vrf> interface gre <gre> ipv6
vrouter running ipv6# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv6 address. |
|------------|------------------|

**link-layer-address (mandatory)**

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

| LINK-LAYER-ADDRESS values | Description      |
|---------------------------|------------------|
| <A.B.C.D>                 | An IPv4 address. |
| <X:X::X:X>                | An IPv6 address. |

### state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface gre <gre> ipv6 neighbor <neighbor> state
```

### network-stack

Network stack parameters for this interface.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack
```

### ipv4

IPv4 parameters.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv4
```

### send-redirects

Send ICMP redirect if host is on the same network than gateway.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv4  
vrouter running ipv4# send-redirects true|false
```

### accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv4  
vrouter running ipv4# accept-redirects true|false
```

### accept-source-route

Accept packets with source route option. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv4
vrouter running ipv4# accept-source-route true|false
```

### arp-announce

Define different restriction levels for announcing the local source IP address from IP packets in ARP requests sent on interface. Increasing the restriction level gives more chance for receiving answer from the resolved target while decreasing the level announces more valid sender’s information.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv4
vrouter running ipv4# arp-announce ARP-ANNOUNCE
```

| ARP-ANNOUNCE values | Description  |
|---------------------|--|
| any                 | Use any local address, configured on any interface.  |
| avoid-not-in-subnet | Try to avoid local addresses that are not in the target’s subnet for this interface. This mode is useful when target hosts reachable via this interface require the source IP address in ARP requests to be part of their logical network configured on the receiving interface. When we generate the request we will check all our subnets that include the target IP and will preserve the source address if it is from such subnet. If there is no such subnet we select source address according to the rules for level 2, ‘best-local’.   |
| best-local          | Always use the best local address for this target. In this mode we ignore the source address in the IP packet and try to select local address that we prefer for talks with the target host. Such local address is selected by looking for primary IP addresses on all our subnets on the outgoing interface that include the target IP address. If no suitable local address is found we select the first local address we have on the outgoing interface or on all other interfaces, with the hope we will receive reply for our request and even sometimes no matter the source IP address we announce. |

### arp-filter

Allows to have multiple network interfaces on the same subnet, and have the ARPs for each interface be answered based on whether or not the kernel would route a packet from the ARP’d IP out that interface (therefore you must use source based routing for this to work). In other words it allows control of which cards (usually 1) will respond to an arp request.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv4
vrouter running ipv4# arp-filter true|false
```

## arp-ignore

Define different modes for sending replies in response to received ARP requests that resolve local target IP addresses.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv4
vrouter running ipv4# arp-ignore ARP-IGNORE
```

| ARP-IGNORE values          | Description  |
|----------------------------|--|
| any                        | Reply for any local target IP address, configured on any interface.  |
| check-interface            | Reply only if the target IP address is local address configured on the incoming interface.   |
| check-interface-and-subnet | Reply only if the target IP address is local address configured on the incoming interface and both with the sender's IP address are part from same subnet on this interface. |
| ignore-scope               | Do not reply for local addresses configured with scope host, only resolutions for global and link addresses are replied.   |
| ignore-all                 | Do not reply for all local addresses.  |

## log-invalid-addresses

Log packets with impossible addresses.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv4
vrouter running ipv4# log-invalid-addresses true|false
```

## ipv6

IPv6 parameters.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv6
```

## autoconfiguration

Autoconfigure addresses using Prefix Information in Router Advertisements.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv6
vrouter running ipv6# autoconfiguration true|false
```

## accept-router-advert

Accept Router Advertisements.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv6
vrouter running ipv6# accept-router-advert ACCEPT-ROUTER-ADVERT
```

| ACCEPT-ROUTER-ADVERT values | Description   |
|-----------------------------|---|
| never                       | Do not accept Router Advertisements.                        |
| norouter-mode               | Accept Router Advertisements if forwarding is disabled.     |
| always                      | Accept Router Advertisements even if forwarding is enabled. |

## accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv6
vrouter running ipv6# accept-redirects true|false
```

## accept-source-route

Accept packets with source route option.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv6
vrouter running ipv6# accept-source-route true|false
```

## router-solicitations

Number of Router Solicitations to send until assuming no routers are present.

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv6
vrouter running ipv6# router-solicitations <int16>
```

### use-temporary-addresses

Preference for Privacy Extensions (RFC4941). Not applied to point-to-point and loopback devices (always 0).

```
vrouter running config# vrf <vrf> interface gre <gre> network-stack ipv6
vrouter running ipv6# use-temporary-addresses USE-TEMPORARY-ADDRESSES
```

| USE-TEMPORARY-ADDRESSES values | Description   |
|--------------------------------|---|
| never                          | Disable Privacy Extensions, i.e. use the public address, subnet prefix/interface id, where interface id is always the same. |
| prefer-public-addresses        | Enable Privacy Extensions, but prefer public addresses over temporary addresses.  |
| always                         | Enable Privacy Extensions and prefer temporary addresses over public addresses.   |

### key

Set the value of the GRE key for this interface.

```
vrouter running config# vrf <vrf> interface gre <gre> key
```

### input

GRE key of incoming packets (overrides the value specified in both).

```
vrouter running config# vrf <vrf> interface gre <gre> key
vrouter running key# input INPUT
```

| INPUT values | Description      |
|--------------|------------------|
| <uint32>     | GRE key type.    |
| <A.B.C.D>    | An IPv4 address. |

### output

GRE key for outgoing packets (overrides the value specified in both).

```
vrouter running config# vrf <vrf> interface gre <gre> key
vrouter running key# output OUTPUT
```

| OUTPUT values | Description      |
|---------------|------------------|
| <uint32>      | GRE key type.    |
| <A.B.C.D>     | An IPv4 address. |

## both

GRE key for incoming and outgoing packets.

```
vrouter running config# vrf <vrf> interface gre <gre> key
vrouter running key# both BOTH
```

| BOTH values | Description      |
|-------------|------------------|
| <uint32>    | GRE key type.    |
| <A.B.C.D>   | An IPv4 address. |

## qos

---

**Note:** requires a Turbo Router Network License.

---

QoS configuration.

```
vrouter running config# vrf <vrf> interface gre <gre> qos
```

## ingress

Ingress QoS configuration.

```
vrouter running config# vrf <vrf> interface gre <gre> qos ingress
```

## rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface gre <gre> qos ingress rate-limit
```

## policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface gre <gre> qos ingress rate-limit
vrouter running rate-limit# policer <leafref>
```

### shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface gre <gre> qos ingress rate-limit
vrouter running rate-limit# shared-policer <leafref>
```

### policer (state only)

Traffic policer.

### bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface gre <gre> qos ingress rate-limit policer
↳ bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface gre <gre> qos ingress rate-limit policer burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface gre <gre> qos ingress rate-limit policer
↳ excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface gre <gre> qos ingress rate-limit policer_
↳excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface gre <gre> qos ingress rate-limit policer_
↳shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface gre <gre> qos ingress rate-limit policer stats_
↳pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface gre <gre> qos ingress rate-limit policer stats_
↳pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface gre <gre> qos ingress rate-limit policer stats_
↳pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface gre <gre> qos ingress rate-limit policer stats_
↳pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface gre <gre> qos ingress rate-limit policer stats_
↳drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface gre <gre> qos ingress rate-limit policer stats_
↳drop-bytes
```

### egress

Egress QoS configuration.

```
vrouter running config# vrf <vrf> interface gre <gre> qos egress
```

## rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface gre <gre> qos egress rate-limit
```

## policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface gre <gre> qos egress rate-limit  
vrouter running rate-limit# policer <leafref>
```

## shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface gre <gre> qos egress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

## policer (state only)

Traffic policer.

## bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface gre <gre> qos egress rate-limit policer  
↳ bandwidth
```

## burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface gre <gre> qos egress rate-limit policer burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface gre <gre> qos egress rate-limit policer excess-  
↳bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface gre <gre> qos egress rate-limit policer excess-  
↳burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface gre <gre> qos egress rate-limit policer shared-  
↳policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface gre <gre> qos egress rate-limit policer stats_  
↳pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface gre <gre> qos egress rate-limit policer stats_
↳pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface gre <gre> qos egress rate-limit policer stats_
↳pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface gre <gre> qos egress rate-limit policer stats_
↳pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface gre <gre> qos egress rate-limit policer stats_
↳drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface gre <gre> qos egress rate-limit policer stats_
↳drop-bytes
```

### counters (state only)

A collection of interface-related statistics objects.

### in-octets (state only)

The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface gre <gre> counters in-octets
```

### in-unicast-pkts (state only)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface gre <gre> counters in-unicast-pkts
```

### in-discards (state only)

The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface gre <gre> counters in-discards
```

### in-errors (state only)

For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface gre <gre> counters in-errors
```

### out-octets (state only)

The total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface gre <gre> counters out-octets
```

### out-unicast-pkts (state only)

The total number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface gre <gre> counters out-unicast-pkts
```

### out-discards (state only)

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface gre <gre> counters out-discards
```

### out-errors (state only)

For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface gre <gre> counters out-errors
```

## ipip

---

**Note:** requires a Turbo Router Network License.

---

The list of ipip interfaces on the device.

```
vrouter running config# vrf <vrf> interface ipip <ipip>
```

|        |                    |
|--------|--------------------|
| <ipip> | An interface name. |
|--------|--------------------|

## mtu

Set the max transmission unit size in octets.

```
vrouter running config# vrf <vrf> interface ipip <ipip>  
vrouter running ipip <ipip># mtu <uint32>
```

## promiscuous

Set promiscuous mode.

```
vrouter running config# vrf <vrf> interface ipip <ipip>  
vrouter running ipip <ipip># promiscuous true|false
```

## description

A textual description of the interface.

```
vrouter running config# vrf <vrf> interface ipip <ipip>  
vrouter running ipip <ipip># description <string>
```

## enabled

The desired (administrative) state of the interface.

```
vrouter running config# vrf <vrf> interface ipip <ipip>  
vrouter running ipip <ipip># enabled true|false
```

## Default value

**true**

### local (mandatory)

The source address that should be used for the tunnel.

```
vrouter running config# vrf <vrf> interface ipip <ipip>
vrouter running ipip <ipip># local LOCAL
```

| LOCAL values | Description      |
|--------------|------------------|
| <A.B.C.D>    | An IPv4 address. |
| <X:X::X:X>   | An IPv6 address. |

### remote (mandatory)

The destination address that should be used for the tunnel.

```
vrouter running config# vrf <vrf> interface ipip <ipip>
vrouter running ipip <ipip># remote REMOTE
```

| REMOTE values | Description      |
|---------------|------------------|
| <A.B.C.D>     | An IPv4 address. |
| <X:X::X:X>    | An IPv6 address. |

### ttl

The time-to-live (or hop limit) that should be utilised for the IP packets used for the tunnel transport.

```
vrouter running config# vrf <vrf> interface ipip <ipip>
vrouter running ipip <ipip># ttl <uint8>
```

### tos

Set the DSCP bits in the Type of Service field.

```
vrouter running config# vrf <vrf> interface ipip <ipip>
vrouter running ipip <ipip># tos <uint8>
```

## link-interface

Route tunneled packets through this interface.

```
vrouter running config# vrf <vrf> interface ipip <ipip>  
vrouter running ipip <ipip># link-interface LINK-INTERFACE
```

|                |                    |
|----------------|--------------------|
| LINK-INTERFACE | An interface name. |
|----------------|--------------------|

## link-vrf

The link vrf name.

```
vrouter running config# vrf <vrf> interface ipip <ipip>  
vrouter running ipip <ipip># link-vrf <string>
```

## ifindex (state only)

System assigned number for each interface. Corresponds to ifIndex object in SNMP Interface MIB.

```
vrouter> show state vrf <vrf> interface ipip <ipip> ifindex
```

## admin-status (state only)

The desired state of the interface. In RFC 7223 this leaf has the same read semantics as ifAdminStatus. Here, it reflects the administrative state as set by enabling or disabling the interface.

```
vrouter> show state vrf <vrf> interface ipip <ipip> admin-status
```

## oper-status (state only)

The current operational state of the interface. This leaf has the same semantics as ifOperStatus.

```
vrouter> show state vrf <vrf> interface ipip <ipip> oper-status
```

## last-change (state only)

This timestamp indicates the time of the last state change of the interface (e.g., up-to-down transition). This corresponds to the ifLastChange object in the standard interface MIB. The value is the timestamp in nanoseconds relative to the Unix Epoch (Jan 1, 1970 00:00:00 UTC).

```
vrouter> show state vrf <vrf> interface ipip <ipip> last-change
```

## ipv4

Parameters for the IPv4 address family.

```
vrouter running config# vrf <vrf> interface ipip <ipip> ipv4
```

## enabled

Controls whether IPv4 is enabled or disabled on this interface. When IPv4 is enabled, this interface is connected to an IPv4 stack, and the interface can send and receive IPv4 packets.

```
vrouter running config# vrf <vrf> interface ipip <ipip> ipv4
vrouter running ipv4# enabled true|false
```

## Default value

true

## address

The list of configured IPv4 addresses on the interface.

```
vrouter running config# vrf <vrf> interface ipip <ipip> ipv4
vrouter running ipv4# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <A.B.C.D>        | An IPv4 address.  |
| <A.B.C.D/M>      | A masked IPv4 address: address and prefix of that subnet. |

**peer**

The IPv4 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv4 address. |
|------|------------------|

**origin (state only)**

The origin of this address, e.g., statically configured, assigned by DHCP, etc..

```
vrouter> show state vrf <vrf> interface ipip <ipip> ipv4 address <address> origin
```

**ipv6**

Parameters for the IPv6 address family.

```
vrouter running config# vrf <vrf> interface ipip <ipip> ipv6
```

**enabled**

Controls whether IPv6 is enabled or disabled on this interface. When IPv6 is enabled, this interface is connected to an IPv6 stack, and the interface can send and receive IPv6 packets.

```
vrouter running config# vrf <vrf> interface ipip <ipip> ipv6
vrouter running ipv6# enabled true|false
```

**Default value**

true

**address**

The list of configured IPv6 addresses on the interface.

```
vrouter running config# vrf <vrf> interface ipip <ipip> ipv6
vrouter running ipv6# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <X:X::X:X>       | An IPv6 address.  |
| <X:X::X:X/M>     | A masked IPv6 address: address and prefix of that subnet. |

### peer

The IPv6 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv6 address. |
|------|------------------|

### origin (state only)

The origin of this address, e.g., static, dhcp, etc.

```
vrouter> show state vrf <vrf> interface ipip <ipip> ipv6 address <address> origin
```

### status (state only)

The status of an address. Most of the states correspond to states from the IPv6 Stateless Address Autoconfiguration protocol.

```
vrouter> show state vrf <vrf> interface ipip <ipip> ipv6 address <address> status
```

### network-stack

Network stack parameters for this interface.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack
```

## ipv4

IPv4 parameters.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv4
```

### send-redirects

Send ICMP redirect if host is on the same network than gateway.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv4  
vrouter running ipv4# send-redirects true|false
```

### accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv4  
vrouter running ipv4# accept-redirects true|false
```

### accept-source-route

Accept packets with source route option. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv4  
vrouter running ipv4# accept-source-route true|false
```

### arp-announce

Define different restriction levels for announcing the local source IP address from IP packets in ARP requests sent on interface. Increasing the restriction level gives more chance for receiving answer from the resolved target while decreasing the level announces more valid sender's information.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv4  
vrouter running ipv4# arp-announce ARP-ANNOUNCE
```

| ARP-AND-Description<br>val-ues | Description  |
|--------------------------------|--|
| any                            | Use any local address, configured on any interface.  |
| avoid-not-in-subnet            | Try to avoid local addresses that are not in the target's subnet for this interface. This mode is useful when target hosts reachable via this interface require the source IP address in ARP requests to be part of their logical network configured on the receiving interface. When we generate the request we will check all our subnets that include the target IP and will preserve the source address if it is from such subnet. If there is no such subnet we select source address according to the rules for level 2, 'best-local'.   |
| best-local                     | Always use the best local address for this target. In this mode we ignore the source address in the IP packet and try to select local address that we prefer for talks with the target host. Such local address is selected by looking for primary IP addresses on all our subnets on the outgoing interface that include the target IP address. If no suitable local address is found we select the first local address we have on the outgoing interface or on all other interfaces, with the hope we will receive reply for our request and even sometimes no matter the source IP address we announce. |

### arp-filter

Allows to have multiple network interfaces on the same subnet, and have the ARPs for each interface be answered based on whether or not the kernel would route a packet from the ARP'd IP out that interface (therefore you must use source based routing for this to work). In other words it allows control of which cards (usually 1) will respond to an arp request.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv4
vrouter running ipv4# arp-filter true|false
```

### arp-ignore

Define different modes for sending replies in response to received ARP requests that resolve local target IP addresses.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv4
vrouter running ipv4# arp-ignore ARP-IGNORE
```

| ARP-IGNORE values          | Description  |
|----------------------------|--|
| any                        | Reply for any local target IP address, configured on any interface.  |
| check-interface            | Reply only if the target IP address is local address configured on the incoming interface.   |
| check-interface-and-subnet | Reply only if the target IP address is local address configured on the incoming interface and both with the sender's IP address are part from same subnet on this interface. |
| ignore-scope               | Do not reply for local addresses configured with scope host, only resolutions for global and link addresses are replied.   |
| ignore-all                 | Do not reply for all local addresses.  |

### log-invalid-addresses

Log packets with impossible addresses.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv4
vrouter running ipv4# log-invalid-addresses true|false
```

### ipv6

IPv6 parameters.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv6
```

### autoconfiguration

Autoconfigure addresses using Prefix Information in Router Advertisements.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv6
vrouter running ipv6# autoconfiguration true|false
```

### accept-router-advert

Accept Router Advertisements.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv6
vrouter running ipv6# accept-router-advert ACCEPT-ROUTER-ADVERT
```

| ACCEPT-ROUTER-ADVERT values | Description   |
|-----------------------------|---|
| never                       | Do not accept Router Advertisements.                        |
| norouter-mode               | Accept Router Advertisements if forwarding is disabled.     |
| always                      | Accept Router Advertisements even if forwarding is enabled. |

### accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv6
vrouter running ipv6# accept-redirects true|false
```

### accept-source-route

Accept packets with source route option.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv6
vrouter running ipv6# accept-source-route true|false
```

### router-solicitations

Number of Router Solicitations to send until assuming no routers are present.

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv6
vrouter running ipv6# router-solicitations <int16>
```

### use-temporary-addresses

Preference for Privacy Extensions (RFC4941). Not applied to point-to-point and loopback devices (always 0).

```
vrouter running config# vrf <vrf> interface ipip <ipip> network-stack ipv6
vrouter running ipv6# use-temporary-addresses USE-TEMPORARY-ADDRESSES
```

| USE-TEMPORARY-ADDRESSES values | Description   |
|--------------------------------|---|
| never                          | Disable Privacy Extensions, i.e. use the public address, subnet prefix/interface id, where interface id is always the same. |
| prefer-public-addresses        | Enable Privacy Extensions, but prefer public addresses over temporary addresses.  |
| always                         | Enable Privacy Extensions and prefer temporary addresses over public addresses.   |

## qos

---

**Note:** requires a Turbo Router Network License.

---

QoS configuration.

```
vrouter running config# vrf <vrf> interface ipip <ipip> qos
```

## ingress

Ingress QoS configuration.

```
vrouter running config# vrf <vrf> interface ipip <ipip> qos ingress
```

## rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface ipip <ipip> qos ingress rate-limit
```

## policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface ipip <ipip> qos ingress rate-limit  
vrouter running rate-limit# policer <leafref>
```

## shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface ipip <ipip> qos ingress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

### **policer (state only)**

Traffic policer.

### **bandwidth (state only)**

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos ingress rate-limit policer_
↳bandwidth
```

### **burst (state only)**

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos ingress rate-limit policer_
↳burst
```

### **excess-bandwidth (state only)**

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos ingress rate-limit policer_
↳excess-bandwidth
```

### **excess-burst (state only)**

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos ingress rate-limit policer_
↳excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos ingress rate-limit policer_
↳shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos ingress rate-limit policer_
↳stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos ingress rate-limit policer_
↳stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos ingress rate-limit policer_
↳stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos ingress rate-limit policer_
↳stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos ingress rate-limit policer.  
↳stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos ingress rate-limit policer.  
↳stats drop-bytes
```

### egress

Egress QoS configuration.

```
vrouter running config# vrf <vrf> interface ipip <ipip> qos egress
```

### rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface ipip <ipip> qos egress rate-limit
```

### policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface ipip <ipip> qos egress rate-limit  
vrouter running rate-limit# policer <leafref>
```

### shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface ipip <ipip> qos egress rate-limit
vrouter running rate-limit# shared-policer <leafref>
```

### policer (state only)

Traffic policer.

### bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos egress rate-limit policer
↳bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos egress rate-limit policer burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos egress rate-limit policer
↳excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos egress rate-limit policer.  
↳excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos egress rate-limit policer.  
↳shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos egress rate-limit policer.  
↳stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface ipip <ipip> qos egress rate-limit policer.  
↳stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouters> show state vrf <vrf> interface ipip <ipip> qos egress rate-limit policer.  
↳stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouters> show state vrf <vrf> interface ipip <ipip> qos egress rate-limit policer.  
↳stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouters> show state vrf <vrf> interface ipip <ipip> qos egress rate-limit policer.  
↳stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouters> show state vrf <vrf> interface ipip <ipip> qos egress rate-limit policer.  
↳stats drop-bytes
```

### counters (state only)

A collection of interface-related statistics objects.

### in-octets (state only)

The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouters> show state vrf <vrf> interface ipip <ipip> counters in-octets
```

### in-unicast-pkts (state only)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface ipip <ipip> counters in-unicast-pkts
```

### in-discards (state only)

The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface ipip <ipip> counters in-discards
```

### in-errors (state only)

For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface ipip <ipip> counters in-errors
```

### out-octets (state only)

The total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface ipip <ipip> counters out-octets
```

### out-unicast-pkts (state only)

The total number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re- initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouters> show state vrf <vrf> interface ipip <ipip> counters out-unicast-pkts
```

### out-discards (state only)

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouters> show state vrf <vrf> interface ipip <ipip> counters out-discards
```

### out-errors (state only)

For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouters> show state vrf <vrf> interface ipip <ipip> counters out-errors
```

### lag

The list of LAG interfaces on the device.

```
vrouters running config# vrf <vrf> interface lag <lag>
```

|       |                    |
|-------|--------------------|
| <lag> | An interface name. |
|-------|--------------------|

## mtu

Set the max transmission unit size in octets.

```
vrouter running config# vrf <vrf> interface lag <lag>  
vrouter running lag <lag># mtu <uint32>
```

## promiscuous

Set promiscuous mode.

```
vrouter running config# vrf <vrf> interface lag <lag>  
vrouter running lag <lag># promiscuous true|false
```

## description

A textual description of the interface.

```
vrouter running config# vrf <vrf> interface lag <lag>  
vrouter running lag <lag># description <string>
```

## enabled

The desired (administrative) state of the interface.

```
vrouter running config# vrf <vrf> interface lag <lag>  
vrouter running lag <lag># enabled true|false
```

## Default value

true

## mode (mandatory)

LAG mode.

```
vrouter running config# vrf <vrf> interface lag <lag>  
vrouter running lag <lag># mode MODE
```

| MODE values   | Description  |
|---------------|--|
| round-robin   | Outgoing traffic is distributed sequentially on each slave.                                  |
| xor           | Outgoing traffic is distributed according to a configurable policy (see policy for details). |
| active-backup | Only one link in the link aggregation will be used at a time.                                |
| lacp          | Full LACP support.   |

### xmit-hash-policy

LAG xmit hash policy to use for slave selection in xor or lacp modes.

```
vrouter running config# vrf <vrf> interface lag <lag>
vrouter running lag <lag># xmit-hash-policy XMIT-HASH-POLICY
```

| XMIT-HASH-POLICY values | Description                        |
|-------------------------|------------------------------------|
| layer2                  | Hash L2 headers.                   |
| layer2+3                | Hash L2 and L3 headers.            |
| layer3+4                | Hash L3 and L4 headers.            |
| encap2+3                | Hash most inner L2 and L3 headers. |
| encap3+4                | Hash most inner L3 and L4 headers. |

### lacp-rate

LACP rate transmission.

```
vrouter running config# vrf <vrf> interface lag <lag>
vrouter running lag <lag># lacp-rate LACP-RATE
```

| LACP-RATE values | Description   |
|------------------|---|
| slow             | In lacp mode, transmit LACPDU packets every 30 seconds. |
| fast             | In lacp mode, transmit LACPDU packets every seconds.    |

### mii-link-monitoring

Define the MII link monitoring frequency in milliseconds.

```
vrouter running config# vrf <vrf> interface lag <lag>
vrouter running lag <lag># mii-link-monitoring <uint32>
```

#### Default value

100

### ifindex (state only)

System assigned number for each interface. Corresponds to ifIndex object in SNMP Interface MIB.

```
vrouter> show state vrf <vrf> interface lag <lag> ifindex
```

### admin-status (state only)

The desired state of the interface. In RFC 7223 this leaf has the same read semantics as ifAdminStatus. Here, it reflects the administrative state as set by enabling or disabling the interface.

```
vrouter> show state vrf <vrf> interface lag <lag> admin-status
```

### oper-status (state only)

The current operational state of the interface. This leaf has the same semantics as ifOperStatus.

```
vrouter> show state vrf <vrf> interface lag <lag> oper-status
```

### last-change (state only)

This timestamp indicates the time of the last state change of the interface (e.g., up-to-down transition). This corresponds to the ifLastChange object in the standard interface MIB. The value is the timestamp in nanoseconds relative to the Unix Epoch (Jan 1, 1970 00:00:00 UTC).

```
vrouter> show state vrf <vrf> interface lag <lag> last-change
```

### ethernet

Top-level container for Ethernet configuration.

```
vrouter running config# vrf <vrf> interface lag <lag> ethernet
```

## mac-address

Assigns a MAC address to the Ethernet interface. If not specified, the corresponding operational state leaf is expected to show the system-assigned MAC address.

```
vrouter running config# vrf <vrf> interface lag <lag> ethernet
vrouter running ethernet# mac-address MAC-ADDRESS
```

|             |  |
|-------------|--|
| MAC-ADDRESS | An IEEE 802 unicast MAC address i.e. the second digit is an even number. Moreover the mac address must not be 00:00:00:00:00:00. |
|-------------|--|

## ipv4

Parameters for the IPv4 address family.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4
```

## enabled

Controls whether IPv4 is enabled or disabled on this interface. When IPv4 is enabled, this interface is connected to an IPv4 stack, and the interface can send and receive IPv4 packets.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4
vrouter running ipv4# enabled true|false
```

## Default value

true

## address

The list of configured IPv4 addresses on the interface.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4
vrouter running ipv4# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <A.B.C.D>        | An IPv4 address.  |
| <A.B.C.D/M>      | A masked IPv4 address: address and prefix of that subnet. |

**peer**

The IPv4 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv4 address. |
|------|------------------|

**origin (state only)**

The origin of this address, e.g., statically configured, assigned by DHCP, etc..

```
vrouter> show state vrf <vrf> interface lag <lag> ipv4 address <address> origin
```

**neighbor**

A list of mappings from IPv4 addresses to link-layer addresses. Entries in this list are used as static entries in the ARP Cache.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4
vrouter running ipv4# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv4 address. |
|------------|------------------|

**link-layer-address (mandatory)**

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

### state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface lag <lag> ipv4 neighbor <neighbor> state
```

### dhcp

DHCP client configuration.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4 dhcp
```

### enabled

Enable or disable DHCP.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4 dhcp  
vrouter running dhcp# enabled true|false
```

### Default value

true

### timeout

Time before deciding that it's not going to be able to contact a server.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4 dhcp  
vrouter running dhcp# timeout <uint32>
```

### Default value

60

### retry

Time before trying again to contact a DHCP server.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4 dhcp  
vrouter running dhcp# retry <uint32>
```

### Default value

300

### select-timeout

Time at which the client stops waiting for other offers from servers.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4 dhcp
vrouter running dhcp# select-timeout <uint32>
```

#### Default value

0

### reboot

Time after trying to reacquire its old address before trying to discover a new address.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4 dhcp
vrouter running dhcp# reboot <uint32>
```

#### Default value

10

### initial-interval

Time between the first attempt to reach a server and the second attempt to reach a server.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4 dhcp
vrouter running dhcp# initial-interval <uint32>
```

#### Default value

10

### dhcp-lease-time

Requested lease time.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4 dhcp
vrouter running dhcp# dhcp-lease-time <uint32>
```

#### Default value

7200

### dhcp-client-identifier-ascii

DHCP client identifier (ASCII).

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4 dhcp  
vrouter running dhcp# dhcp-client-identifier-ascii <string>
```

### dhcp-client-identifier-hexa

DHCP client identifier (hexadecimal).

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4 dhcp  
vrouter running dhcp# dhcp-client-identifier-hexa <string>
```

### host-name

DHCP client name.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4 dhcp  
vrouter running dhcp# host-name <string>
```

### request

DHCP requests.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv4 dhcp  
vrouter running dhcp# request REQUEST
```

| REQUEST values       | Description  |
|----------------------|--|
| subnet-mask          | Client's subnet mask.  |
| broadcast-address    | Broadcast address in use on the client's subnet.                     |
| time-offset          | Offset of the client's subnet in seconds from UTC.                   |
| routers              | List of IP addresses for routers on the client's subnet.             |
| domain-name          | Domain name used when resolving hostnames with DNS.                  |
| domain-search        | Domain search list used when resolving hostnames with DNS.           |
| domain-name-servers  | List of DNS name servers available to the client.                    |
| host-name            | Name of the client.  |
| nis-domain           | Name of the client's NIS (Sun Network Information Services) domain.  |
| nis-servers          | List of IP addresses indicating NIS servers available to the client. |
| ntp-servers          | List of IP addresses indicating NTP servers available to the client. |
| interface-mtu        | MTU to use on this interface.  |
| netbios-name-servers | List of RFC 1001/1002 NBNS name servers.                             |
| netbios-scope        | NetBIOS over TCP/IP scope parameter for the client.                  |

### Default value

subnet-mask  
 broadcast-address  
 time-offset  
 routers  
 domain-name  
 domain-search  
 domain-name-servers  
 host-name  
 nis-domain  
 nis-servers  
 ntp-servers  
 interface-mtu

### current-lease (state only)

Current lease.

### fixed-address (state only)

The IPv4 address on the interface.

```
vrouter> show state vrf <vrf> interface lag <lag> ipv4 dhcp current-lease fixed-address
```

### renew (state only)

Time at which the client should begin trying to contact its server to renew its lease.

```
vrouter> show state vrf <vrf> interface lag <lag> ipv4 dhcp current-lease renew
```

### rebind (state only)

Time at which the client should begin to try to contact any dhcp server to renew its lease.

```
vrouter> show state vrf <vrf> interface lag <lag> ipv4 dhcp current-lease rebind
```

### expire (state only)

Time at which the client must stop using a lease if it has not been able to renew it.

```
vrouter> show state vrf <vrf> interface lag <lag> ipv4 dhcp current-lease expire
```

## ipv6

Parameters for the IPv6 address family.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv6
```

### enabled

Controls whether IPv6 is enabled or disabled on this interface. When IPv6 is enabled, this interface is connected to an IPv6 stack, and the interface can send and receive IPv6 packets.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv6  
vrouter running ipv6# enabled true|false
```

### Default value

true

## address

The list of configured IPv6 addresses on the interface.

```
vrouters running config# vrf <vrf> interface lag <lag> ipv6
vrouters running ipv6# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <X:X::X:X>       | An IPv6 address.  |
| <X:X::X:X/M>     | A masked IPv6 address: address and prefix of that subnet. |

## peer

The IPv6 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv6 address. |
|------|------------------|

## origin (state only)

The origin of this address, e.g., static, dhcp, etc.

```
vrouters> show state vrf <vrf> interface lag <lag> ipv6 address <address> origin
```

## status (state only)

The status of an address. Most of the states correspond to states from the IPv6 Stateless Address Autoconfiguration protocol.

```
vrouters> show state vrf <vrf> interface lag <lag> ipv6 address <address> status
```

## neighbor

List of IPv6 neighbors.

```
vrouters running config# vrf <vrf> interface lag <lag> ipv6
vrouters running ipv6# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv6 address. |
|------------|------------------|

### link-layer-address (mandatory)

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

### router (state only)

Indicates that the neighbor node acts as a router.

```
vrouter> show state vrf <vrf> interface lag <lag> ipv6 neighbor <neighbor> router
```

### state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface lag <lag> ipv6 neighbor <neighbor> state
```

### network-stack

Network stack parameters for this interface.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack
```

### ipv4

IPv4 parameters.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv4
```

## send-redirects

Send ICMP redirect if host is on the same network than gateway.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv4  
vrouter running ipv4# send-redirects true|false
```

## accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv4  
vrouter running ipv4# accept-redirects true|false
```

## accept-source-route

Accept packets with source route option. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv4  
vrouter running ipv4# accept-source-route true|false
```

## arp-announce

Define different restriction levels for announcing the local source IP address from IP packets in ARP requests sent on interface. Increasing the restriction level gives more chance for receiving answer from the resolved target while decreasing the level announces more valid sender's information.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv4  
vrouter running ipv4# arp-announce ARP-ANNOUNCE
```

| ARP-AND-Description<br>val-ues | Description  |
|--------------------------------|--|
| any                            | Use any local address, configured on any interface.  |
| avoid-not-in-subnet            | Try to avoid local addresses that are not in the target's subnet for this interface. This mode is useful when target hosts reachable via this interface require the source IP address in ARP requests to be part of their logical network configured on the receiving interface. When we generate the request we will check all our subnets that include the target IP and will preserve the source address if it is from such subnet. If there is no such subnet we select source address according to the rules for level 2, 'best-local'.   |
| best-local                     | Always use the best local address for this target. In this mode we ignore the source address in the IP packet and try to select local address that we prefer for talks with the target host. Such local address is selected by looking for primary IP addresses on all our subnets on the outgoing interface that include the target IP address. If no suitable local address is found we select the first local address we have on the outgoing interface or on all other interfaces, with the hope we will receive reply for our request and even sometimes no matter the source IP address we announce. |

### arp-filter

Allows to have multiple network interfaces on the same subnet, and have the ARPs for each interface be answered based on whether or not the kernel would route a packet from the ARP'd IP out that interface (therefore you must use source based routing for this to work). In other words it allows control of which cards (usually 1) will respond to an arp request.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv4
vrouter running ipv4# arp-filter true|false
```

### arp-ignore

Define different modes for sending replies in response to received ARP requests that resolve local target IP addresses.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv4
vrouter running ipv4# arp-ignore ARP-IGNORE
```

| ARP-IGNORE values          | Description  |
|----------------------------|--|
| any                        | Reply for any local target IP address, configured on any interface.  |
| check-interface            | Reply only if the target IP address is local address configured on the incoming interface.   |
| check-interface-and-subnet | Reply only if the target IP address is local address configured on the incoming interface and both with the sender's IP address are part from same subnet on this interface. |
| ignore-scope               | Do not reply for local addresses configured with scope host, only resolutions for global and link addresses are replied.   |
| ignore-all                 | Do not reply for all local addresses.  |

### log-invalid-addresses

Log packets with impossible addresses.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv4
vrouter running ipv4# log-invalid-addresses true|false
```

### ipv6

IPv6 parameters.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv6
```

### autoconfiguration

Autoconfigure addresses using Prefix Information in Router Advertisements.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv6
vrouter running ipv6# autoconfiguration true|false
```

### accept-router-advert

Accept Router Advertisements.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv6
vrouter running ipv6# accept-router-advert ACCEPT-ROUTER-ADVERT
```

| ACCEPT-ROUTER-ADVERT values | Description   |
|-----------------------------|---|
| never                       | Do not accept Router Advertisements.                        |
| norouter-mode               | Accept Router Advertisements if forwarding is disabled.     |
| always                      | Accept Router Advertisements even if forwarding is enabled. |

### accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv6
vrouter running ipv6# accept-redirects true|false
```

### accept-source-route

Accept packets with source route option.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv6
vrouter running ipv6# accept-source-route true|false
```

### router-solicitations

Number of Router Solicitations to send until assuming no routers are present.

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv6
vrouter running ipv6# router-solicitations <int16>
```

### use-temporary-addresses

Preference for Privacy Extensions (RFC4941). Not applied to point-to-point and loopback devices (always 0).

```
vrouter running config# vrf <vrf> interface lag <lag> network-stack ipv6
vrouter running ipv6# use-temporary-addresses USE-TEMPORARY-ADDRESSES
```

| USE-TEMPORARY-ADDRESSES values | Description   |
|--------------------------------|---|
| never                          | Disable Privacy Extensions, i.e. use the public address, subnet prefix/interface id, where interface id is always the same. |
| prefer-public-addresses        | Enable Privacy Extensions, but prefer public addresses over temporary addresses.  |
| always                         | Enable Privacy Extensions and prefer temporary addresses over public addresses.   |

## link-interface

Set this interface as slave of this LAG.

```
vrouter running config# vrf <vrf> interface lag <lag>  
vrouter running lag <lag># link-interface <link-interface>
```

|                  |                    |
|------------------|--------------------|
| <link-interface> | An interface name. |
|------------------|--------------------|

## state (state only)

Slave state.

```
vrouter> show state vrf <vrf> interface lag <lag> link-interface <link-interface> state
```

## link (state only)

Slave MII link monitoring status.

```
vrouter> show state vrf <vrf> interface lag <lag> link-interface <link-interface> link
```

## failure-count (state only)

Slave failure count.

```
vrouter> show state vrf <vrf> interface lag <lag> link-interface <link-interface> ↵  
↵ failure-count
```

## primary

Configure primary interface for the active-backup mode.

```
vrouter running config# vrf <vrf> interface lag <lag>  
vrouter running lag <lag># primary interface INTERFACE reselect-policy RESELECT-POLICY
```

### interface (mandatory)

Lag primary interface. After recovery, this interface become the active interface according to the reselect policy.

```
interface INTERFACE
```

|           |                    |
|-----------|--------------------|
| INTERFACE | An interface name. |
|-----------|--------------------|

### reselect-policy

Specifies the reselection policy for the primary interface. This affects how the primary interface is chosen to become active when failure of the current active interface or recovery of the primary interface occurs.

```
reselect-policy RESELECT-POLICY
```

| RESELECT-POLICY values | Description  |
|------------------------|--|
| always                 | The primary interface becomes active whenever it comes back up.  |
| better                 | The primary interface becomes active when it comes back up, if its speed and duplex is better than the speed and duplex of the current active slave. |
| failure                | The primary interface becomes active only if it is up and the current active interface fails.  |

#### Default value

always

### qos

---

**Note:** requires a Turbo Router Network License.

---

QoS configuration.

```
vrouter running config# vrf <vrf> interface lag <lag> qos
```

## ingress

Ingress QoS configuration.

```
vrouter running config# vrf <vrf> interface lag <lag> qos ingress
```

## rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface lag <lag> qos ingress rate-limit
```

## policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface lag <lag> qos ingress rate-limit  
vrouter running rate-limit# policer <leafref>
```

## shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface lag <lag> qos ingress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

## policer (state only)

Traffic policer.

## bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface lag <lag> qos ingress rate-limit policer ↵  
↵bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface lag <lag> qos ingress rate-limit policer burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface lag <lag> qos ingress rate-limit policer_
↳excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface lag <lag> qos ingress rate-limit policer_
↳excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface lag <lag> qos ingress rate-limit policer_
↳shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface lag <lag> qos ingress rate-limit policer stats_
↳pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface lag <lag> qos ingress rate-limit policer stats_
↳pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface lag <lag> qos ingress rate-limit policer stats_
↳pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface lag <lag> qos ingress rate-limit policer stats_
↳pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface lag <lag> qos ingress rate-limit policer stats_
↳drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface lag <lag> qos ingress rate-limit policer stats_
↳ drop-bytes
```

### egress

Egress QoS configuration.

```
vrouter running config# vrf <vrf> interface lag <lag> qos egress
```

### rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface lag <lag> qos egress rate-limit
```

### policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface lag <lag> qos egress rate-limit
vrouter running rate-limit# policer <leafref>
```

### shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface lag <lag> qos egress rate-limit
vrouter running rate-limit# shared-policer <leafref>
```

### policer (state only)

Traffic policer.

### bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface lag <lag> qos egress rate-limit policer ↵  
↵bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface lag <lag> qos egress rate-limit policer burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface lag <lag> qos egress rate-limit policer excess-  
↵bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface lag <lag> qos egress rate-limit policer excess-  
↵burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface lag <lag> qos egress rate-limit policer shared-  
↳policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouter> show state vrf <vrf> interface lag <lag> qos egress rate-limit policer stats_  
↳pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouter> show state vrf <vrf> interface lag <lag> qos egress rate-limit policer stats_  
↳pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouter> show state vrf <vrf> interface lag <lag> qos egress rate-limit policer stats_  
↳pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouter> show state vrf <vrf> interface lag <lag> qos egress rate-limit policer stats_  
↳pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface lag <lag> qos egress rate-limit policer stats_
↳ drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface lag <lag> qos egress rate-limit policer stats_
↳ drop-bytes
```

### counters (state only)

A collection of interface-related statistics objects.

### in-octets (state only)

The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface lag <lag> counters in-octets
```

### in-unicast-pkts (state only)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface lag <lag> counters in-unicast-pkts
```

### in-discards (state only)

The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface lag <lag> counters in-discards
```

### in-errors (state only)

For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface lag <lag> counters in-errors
```

### out-octets (state only)

The total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface lag <lag> counters out-octets
```

### out-unicast-pkts (state only)

The total number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface lag <lag> counters out-unicast-pkts
```

### out-discards (state only)

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface lag <lag> counters out-discards
```

### out-errors (state only)

For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface lag <lag> counters out-errors
```

### loopback

---

**Note:** requires a Turbo Router Network License.

---

The list of loopback interfaces on the device.

```
vrouter running config# vrf <vrf> interface loopback <loopback>
```

|            |                    |
|------------|--------------------|
| <loopback> | An interface name. |
|------------|--------------------|

### mtu

Set the max transmission unit size in octets.

```
vrouter running config# vrf <vrf> interface loopback <loopback>  
vrouter running loopback <loopback># mtu <uint32>
```

## promiscuous

Set promiscuous mode.

```
vrouter running config# vrf <vrf> interface loopback <loopback>  
vrouter running loopback <loopback># promiscuous true|false
```

## description

A textual description of the interface.

```
vrouter running config# vrf <vrf> interface loopback <loopback>  
vrouter running loopback <loopback># description <string>
```

## enabled

The desired (administrative) state of the interface.

```
vrouter running config# vrf <vrf> interface loopback <loopback>  
vrouter running loopback <loopback># enabled true|false
```

## Default value

true

## ifindex (state only)

System assigned number for each interface. Corresponds to ifIndex object in SNMP Interface MIB.

```
vrouter> show state vrf <vrf> interface loopback <loopback> ifindex
```

## admin-status (state only)

The desired state of the interface. In RFC 7223 this leaf has the same read semantics as ifAdminStatus. Here, it reflects the administrative state as set by enabling or disabling the interface.

```
vrouter> show state vrf <vrf> interface loopback <loopback> admin-status
```

### oper-status (state only)

The current operational state of the interface. This leaf has the same semantics as ifOperStatus.

```
vrouter> show state vrf <vrf> interface loopback <loopback> oper-status
```

### last-change (state only)

This timestamp indicates the time of the last state change of the interface (e.g., up-to-down transition). This corresponds to the ifLastChange object in the standard interface MIB. The value is the timestamp in nanoseconds relative to the Unix Epoch (Jan 1, 1970 00:00:00 UTC).

```
vrouter> show state vrf <vrf> interface loopback <loopback> last-change
```

### ipv4

Parameters for the IPv4 address family.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4
```

### enabled

Controls whether IPv4 is enabled or disabled on this interface. When IPv4 is enabled, this interface is connected to an IPv4 stack, and the interface can send and receive IPv4 packets.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4
vrouter running ipv4# enabled true|false
```

### Default value

true

### address

The list of configured IPv4 addresses on the interface.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4
vrouter running ipv4# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <A.B.C.D>        | An IPv4 address.  |
| <A.B.C.D/M>      | A masked IPv4 address: address and prefix of that subnet. |

**peer**

The IPv4 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv4 address. |
|------|------------------|

**origin (state only)**

The origin of this address, e.g., statically configured, assigned by DHCP, etc..

```
vrouter> show state vrf <vrf> interface loopback <loopback> ipv4 address <address>
↪origin
```

**neighbor**

A list of mappings from IPv4 addresses to link-layer addresses. Entries in this list are used as static entries in the ARP Cache.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4
vrouter running ipv4# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv4 address. |
|------------|------------------|

**link-layer-address (mandatory)**

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

## state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface loopback <loopback> ipv4 neighbor <neighbor>_
↪state
```

## dhcp

DHCP client configuration.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4 dhcp
```

## enabled

Enable or disable DHCP.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4 dhcp
vrouter running dhcp# enabled true|false
```

## Default value

true

## timeout

Time before deciding that it's not going to be able to contact a server.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4 dhcp
vrouter running dhcp# timeout <uint32>
```

## Default value

60

## retry

Time before trying again to contact a DHCP server.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4 dhcp
vrouter running dhcp# retry <uint32>
```

## Default value

300

### select-timeout

Time at which the client stops waiting for other offers from servers.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4 dhcp
vrouter running dhcp# select-timeout <uint32>
```

#### Default value

0

### reboot

Time after trying to reacquire its old address before trying to discover a new address.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4 dhcp
vrouter running dhcp# reboot <uint32>
```

#### Default value

10

### initial-interval

Time between the first attempt to reach a server and the second attempt to reach a server.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4 dhcp
vrouter running dhcp# initial-interval <uint32>
```

#### Default value

10

### dhcp-lease-time

Requested lease time.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4 dhcp
vrouter running dhcp# dhcp-lease-time <uint32>
```

#### Default value

7200

### dhcp-client-identifier-ascii

DHCP client identifier (ASCII).

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4 dhcp  
vrouter running dhcp# dhcp-client-identifier-ascii <string>
```

### dhcp-client-identifier-hexa

DHCP client identifier (hexadecimal).

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4 dhcp  
vrouter running dhcp# dhcp-client-identifier-hexa <string>
```

### host-name

DHCP client name.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4 dhcp  
vrouter running dhcp# host-name <string>
```

### request

DHCP requests.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv4 dhcp  
vrouter running dhcp# request REQUEST
```

| REQUEST values       | Description  |
|----------------------|--|
| subnet-mask          | Client's subnet mask.  |
| broadcast-address    | Broadcast address in use on the client's subnet.                     |
| time-offset          | Offset of the client's subnet in seconds from UTC.                   |
| routers              | List of IP addresses for routers on the client's subnet.             |
| domain-name          | Domain name used when resolving hostnames with DNS.                  |
| domain-search        | Domain search list used when resolving hostnames with DNS.           |
| domain-name-servers  | List of DNS name servers available to the client.                    |
| host-name            | Name of the client.  |
| nis-domain           | Name of the client's NIS (Sun Network Information Services) domain.  |
| nis-servers          | List of IP addresses indicating NIS servers available to the client. |
| ntp-servers          | List of IP addresses indicating NTP servers available to the client. |
| interface-mtu        | MTU to use on this interface.  |
| netbios-name-servers | List of RFC 1001/1002 NBNS name servers.                             |
| netbios-scope        | NetBIOS over TCP/IP scope parameter for the client.                  |

### Default value

subnet-mask  
 broadcast-address  
 time-offset  
 routers  
 domain-name  
 domain-search  
 domain-name-servers  
 host-name  
 nis-domain  
 nis-servers  
 ntp-servers  
 interface-mtu

### current-lease (state only)

Current lease.

### fixed-address (state only)

The IPv4 address on the interface.

```
vrouter> show state vrf <vrf> interface loopback <loopback> ipv4 dhcp current-lease_
↳fixed-address
```

### renew (state only)

Time at which the client should begin trying to contact its server to renew its lease.

```
vrouter> show state vrf <vrf> interface loopback <loopback> ipv4 dhcp current-lease_
↳renew
```

### rebind (state only)

Time at which the client should begin to try to contact any dhcp server to renew its lease.

```
vrouter> show state vrf <vrf> interface loopback <loopback> ipv4 dhcp current-lease_
↳rebind
```

### expire (state only)

Time at which the client must stop using a lease if it has not been able to renew it.

```
vrouter> show state vrf <vrf> interface loopback <loopback> ipv4 dhcp current-lease_
↳expire
```

### ipv6

Parameters for the IPv6 address family.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv6
```

**enabled**

Controls whether IPv6 is enabled or disabled on this interface. When IPv6 is enabled, this interface is connected to an IPv6 stack, and the interface can send and receive IPv6 packets.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv6
vrouter running ipv6# enabled true|false
```

**Default value**

true

**address**

The list of configured IPv6 addresses on the interface.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv6
vrouter running ipv6# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <X:X::X:X>       | An IPv6 address.  |
| <X:X::X:X/M>     | A masked IPv6 address: address and prefix of that subnet. |

**peer**

The IPv6 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv6 address. |
|------|------------------|

**origin (state only)**

The origin of this address, e.g., static, dhcp, etc.

```
vrouter> show state vrf <vrf> interface loopback <loopback> ipv6 address <address>
↪origin
```

### status (state only)

The status of an address. Most of the states correspond to states from the IPv6 Stateless Address Autoconfiguration protocol.

```
vrouter> show state vrf <vrf> interface loopback <loopback> ipv6 address <address>↵
↵status
```

### neighbor

List of IPv6 neighbors.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ipv6
vrouter running ipv6# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv6 address. |
|------------|------------------|

### link-layer-address (mandatory)

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

### router (state only)

Indicates that the neighbor node acts as a router.

```
vrouter> show state vrf <vrf> interface loopback <loopback> ipv6 neighbor <neighbor>↵
↵router
```

### state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface loopback <loopback> ipv6 neighbor <neighbor>↵
↵state
```

## network-stack

Network stack parameters for this interface.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack
```

## ipv4

IPv4 parameters.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv4
```

## send-redirects

Send ICMP redirect if host is on the same network than gateway.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv4  
vrouter running ipv4# send-redirects true|false
```

## accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv4  
vrouter running ipv4# accept-redirects true|false
```

## accept-source-route

Accept packets with source route option. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv4  
vrouter running ipv4# accept-source-route true|false
```

### arp-announce

Define different restriction levels for announcing the local source IP address from IP packets in ARP requests sent on interface. Increasing the restriction level gives more chance for receiving answer from the resolved target while decreasing the level announces more valid sender’s information.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv4
vrouter running ipv4# arp-announce ARP-ANNOUNCE
```

| ARP-ANNOUNCE values | Description  |
|---------------------|--|
| any                 | Use any local address, configured on any interface.  |
| avoid-not-in-subnet | Try to avoid local addresses that are not in the target’s subnet for this interface. This mode is useful when target hosts reachable via this interface require the source IP address in ARP requests to be part of their logical network configured on the receiving interface. When we generate the request we will check all our subnets that include the target IP and will preserve the source address if it is from such subnet. If there is no such subnet we select source address according to the rules for level 2, ‘best-local’.   |
| best-local          | Always use the best local address for this target. In this mode we ignore the source address in the IP packet and try to select local address that we prefer for talks with the target host. Such local address is selected by looking for primary IP addresses on all our subnets on the outgoing interface that include the target IP address. If no suitable local address is found we select the first local address we have on the outgoing interface or on all other interfaces, with the hope we will receive reply for our request and even sometimes no matter the source IP address we announce. |

### arp-filter

Allows to have multiple network interfaces on the same subnet, and have the ARPs for each interface be answered based on whether or not the kernel would route a packet from the ARP’d IP out that interface (therefore you must use source based routing for this to work). In other words it allows control of which cards (usually 1) will respond to an arp request.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv4
vrouter running ipv4# arp-filter true|false
```

## arp-ignore

Define different modes for sending replies in response to received ARP requests that resolve local target IP addresses.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv4
vrouter running ipv4# arp-ignore ARP-IGNORE
```

| ARP-IGNORE values          | Description  |
|----------------------------|--|
| any                        | Reply for any local target IP address, configured on any interface.  |
| check-interface            | Reply only if the target IP address is local address configured on the incoming interface.   |
| check-interface-and-subnet | Reply only if the target IP address is local address configured on the incoming interface and both with the sender's IP address are part from same subnet on this interface. |
| ignore-scope               | Do not reply for local addresses configured with scope host, only resolutions for global and link addresses are replied.   |
| ignore-all                 | Do not reply for all local addresses.  |

## log-invalid-addresses

Log packets with impossible addresses.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv4
vrouter running ipv4# log-invalid-addresses true|false
```

## ipv6

IPv6 parameters.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv6
```

## autoconfiguration

Autoconfigure addresses using Prefix Information in Router Advertisements.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv6
vrouter running ipv6# autoconfiguration true|false
```

## accept-router-advert

Accept Router Advertisements.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv6
vrouter running ipv6# accept-router-advert ACCEPT-ROUTER-ADVERT
```

| ACCEPT-ROUTER-ADVERT values | Description   |
|-----------------------------|---|
| never                       | Do not accept Router Advertisements.                        |
| norouter-mode               | Accept Router Advertisements if forwarding is disabled.     |
| always                      | Accept Router Advertisements even if forwarding is enabled. |

## accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv6
vrouter running ipv6# accept-redirects true|false
```

## accept-source-route

Accept packets with source route option.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv6
vrouter running ipv6# accept-source-route true|false
```

## router-solicitations

Number of Router Solicitations to send until assuming no routers are present.

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv6
vrouter running ipv6# router-solicitations <int16>
```

## use-temporary-addresses

Preference for Privacy Extensions (RFC4941). Not applied to point-to-point and loopback devices (always 0).

```
vrouter running config# vrf <vrf> interface loopback <loopback> network-stack ipv6
vrouter running ipv6# use-temporary-addresses USE-TEMPORARY-ADDRESSES
```

| USE-TEMPORARY-ADDRESSES values | Description   |
|--------------------------------|---|
| never                          | Disable Privacy Extensions, i.e. use the public address, subnet prefix/interface id, where interface id is always the same. |
| prefer-public-addresses        | Enable Privacy Extensions, but prefer public addresses over temporary addresses.  |
| always                         | Enable Privacy Extensions and prefer temporary addresses over public addresses.   |

## ethernet

Top-level container for Ethernet configuration.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ethernet
```

## mac-address

Assigns a MAC address to the Ethernet interface. If not specified, the corresponding operational state leaf is expected to show the system-assigned MAC address.

```
vrouter running config# vrf <vrf> interface loopback <loopback> ethernet
vrouter running ethernet# mac-address MAC-ADDRESS
```

|             |  |
|-------------|--|
| MAC-ADDRESS | An IEEE 802 unicast MAC address i.e. the second digit is an even number. Moreover the mac address must not be 00:00:00:00:00:00. |
|-------------|--|

## qos

---

**Note:** requires a Turbo Router Network License.

---

QoS configuration.

```
vrouter running config# vrf <vrf> interface loopback <loopback> qos
```

## ingress

Ingress QoS configuration.

```
vrouter running config# vrf <vrf> interface loopback <loopback> qos ingress
```

## rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface loopback <loopback> qos ingress rate-limit
```

## policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface loopback <loopback> qos ingress rate-limit  
vrouter running rate-limit# policer <leafref>
```

## shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface loopback <loopback> qos ingress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

## policer (state only)

Traffic policer.

## bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos ingress rate-limit  
↳ policer bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos ingress rate-limit  
↳policer burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos ingress rate-limit  
↳policer excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos ingress rate-limit  
↳policer excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos ingress rate-limit  
↳policer shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos ingress rate-limit.  
↳policer stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos ingress rate-limit.  
↳policer stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos ingress rate-limit.  
↳policer stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos ingress rate-limit.  
↳policer stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos ingress rate-limit.  
↳policer stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos ingress rate-limit.  
↳ policer stats drop-bytes
```

### egress

Egress QoS configuration.

```
vrouter running config# vrf <vrf> interface loopback <loopback> qos egress
```

### rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface loopback <loopback> qos egress rate-limit
```

### policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface loopback <loopback> qos egress rate-limit  
vrouter running rate-limit# policer <leafref>
```

### shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface loopback <loopback> qos egress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

### **policer (state only)**

Traffic policer.

### **bandwidth (state only)**

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos egress rate-limit.  
↳policer bandwidth
```

### **burst (state only)**

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos egress rate-limit.  
↳policer burst
```

### **excess-bandwidth (state only)**

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos egress rate-limit.  
↳policer excess-bandwidth
```

### **excess-burst (state only)**

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos egress rate-limit.  
↳policer excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos egress rate-limit.  
↳policer shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos egress rate-limit.  
↳policer stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos egress rate-limit.  
↳policer stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos egress rate-limit.  
↳policer stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos egress rate-limit.  
↳policer stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos egress rate-limit.  
↳ policer stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface loopback <loopback> qos egress rate-limit.  
↳ policer stats drop-bytes
```

### counters (state only)

A collection of interface-related statistics objects.

### in-octets (state only)

The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface loopback <loopback> counters in-octets
```

### in-unicast-pkts (state only)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface loopback <loopback> counters in-unicast-pkts
```

### in-discards (state only)

The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface loopback <loopback> counters in-discards
```

### in-errors (state only)

For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface loopback <loopback> counters in-errors
```

### out-octets (state only)

The total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface loopback <loopback> counters out-octets
```

### out-unicast-pkts (state only)

The total number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface loopback <loopback> counters out-unicast-pkts
```

### out-discards (state only)

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouters> show state vrf <vrf> interface loopback <loopback> counters out-discards
```

### out-errors (state only)

For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouters> show state vrf <vrf> interface loopback <loopback> counters out-errors
```

### physical

The list of physical interfaces on the device.

```
vrouters running config# vrf <vrf> interface physical <physical>
```

|            |                    |
|------------|--------------------|
| <physical> | An interface name. |
|------------|--------------------|

### port (mandatory)

Reference to a physical network port.

```
vrouters running config# vrf <vrf> interface physical <physical>
vrouters running physical <physical># port PORT
```

| PORT values        | Description            |
|--------------------|------------------------|
| <pci-port>         | PCI port name.         |
| <device-tree-port> | Device tree port name. |
| <device-tree-port> | Hyper-V port name.     |

## mtu

Set the max transmission unit size in octets.

```
vrouter running config# vrf <vrf> interface physical <physical>  
vrouter running physical <physical># mtu <uint32>
```

## promiscuous

Set promiscuous mode.

```
vrouter running config# vrf <vrf> interface physical <physical>  
vrouter running physical <physical># promiscuous true|false
```

## description

A textual description of the interface.

```
vrouter running config# vrf <vrf> interface physical <physical>  
vrouter running physical <physical># description <string>
```

## enabled

The desired (administrative) state of the interface.

```
vrouter running config# vrf <vrf> interface physical <physical>  
vrouter running physical <physical># enabled true|false
```

## Default value

true

## rx-cp-protection

Enable Rx Control Plane Protection.

```
vrouter running config# vrf <vrf> interface physical <physical>  
vrouter running physical <physical># rx-cp-protection true|false
```

## tx-cp-protection

Enable Tx Control Plane Protection.

```
vrouter running config# vrf <vrf> interface physical <physical>  
vrouter running physical <physical># tx-cp-protection true|false
```

## ifindex (state only)

System assigned number for each interface. Corresponds to ifIndex object in SNMP Interface MIB.

```
vrouter> show state vrf <vrf> interface physical <physical> ifindex
```

## admin-status (state only)

The desired state of the interface. In RFC 7223 this leaf has the same read semantics as ifAdminStatus. Here, it reflects the administrative state as set by enabling or disabling the interface.

```
vrouter> show state vrf <vrf> interface physical <physical> admin-status
```

## oper-status (state only)

The current operational state of the interface. This leaf has the same semantics as ifOperStatus.

```
vrouter> show state vrf <vrf> interface physical <physical> oper-status
```

## last-change (state only)

This timestamp indicates the time of the last state change of the interface (e.g., up-to-down transition). This corresponds to the ifLastChange object in the standard interface MIB. The value is the timestamp in nanoseconds relative to the Unix Epoch (Jan 1, 1970 00:00:00 UTC).

```
vrouter> show state vrf <vrf> interface physical <physical> last-change
```

## ipv4

Parameters for the IPv4 address family.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4
```

### enabled

Controls whether IPv4 is enabled or disabled on this interface. When IPv4 is enabled, this interface is connected to an IPv4 stack, and the interface can send and receive IPv4 packets.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4
vrouter running ipv4# enabled true|false
```

### Default value

true

### address

The list of configured IPv4 addresses on the interface.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4
vrouter running ipv4# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <A.B.C.D>        | An IPv4 address.  |
| <A.B.C.D/M>      | A masked IPv4 address: address and prefix of that subnet. |

### peer

The IPv4 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv4 address. |
|------|------------------|

### origin (state only)

The origin of this address, e.g., statically configured, assigned by DHCP, etc..

```
vrouter> show state vrf <vrf> interface physical <physical> ipv4 address <address>_
↳origin
```

### neighbor

A list of mappings from IPv4 addresses to link-layer addresses. Entries in this list are used as static entries in the ARP Cache.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4
vrouter running ipv4# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv4 address. |
|------------|------------------|

### link-layer-address (mandatory)

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

### state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface physical <physical> ipv4 neighbor <neighbor>_
↳state
```

### dhcp

DHCP client configuration.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4 dhcp
```

## enabled

Enable or disable DHCP.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4 dhcp
vrouter running dhcp# enabled true|false
```

### Default value

true

## timeout

Time before deciding that it's not going to be able to contact a server.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4 dhcp
vrouter running dhcp# timeout <uint32>
```

### Default value

60

## retry

Time before trying again to contact a DHCP server.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4 dhcp
vrouter running dhcp# retry <uint32>
```

### Default value

300

## select-timeout

Time at which the client stops waiting for other offers from servers.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4 dhcp
vrouter running dhcp# select-timeout <uint32>
```

### Default value

0

## reboot

Time after trying to reacquire its old address before trying to discover a new address.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4 dhcp
vrouter running dhcp# reboot <uint32>
```

### Default value

10

## initial-interval

Time between the first attempt to reach a server and the second attempt to reach a server.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4 dhcp
vrouter running dhcp# initial-interval <uint32>
```

### Default value

10

## dhcp-lease-time

Requested lease time.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4 dhcp
vrouter running dhcp# dhcp-lease-time <uint32>
```

### Default value

7200

## dhcp-client-identifier-ascii

DHCP client identifier (ASCII).

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4 dhcp
vrouter running dhcp# dhcp-client-identifier-ascii <string>
```

## dhcp-client-identifier-hexa

DHCP client identifier (hexadecimal).

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4 dhcp
vrouter running dhcp# dhcp-client-identifier-hexa <string>
```

## host-name

DHCP client name.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4 dhcp
vrouter running dhcp# host-name <string>
```

## request

DHCP requests.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv4 dhcp
vrouter running dhcp# request REQUEST
```

| REQUEST values       | Description  |
|----------------------|--|
| subnet-mask          | Client's subnet mask.  |
| broadcast-address    | Broadcast address in use on the client's subnet.                     |
| time-offset          | Offset of the client's subnet in seconds from UTC.                   |
| routers              | List of IP addresses for routers on the client's subnet.             |
| domain-name          | Domain name used when resolving hostnames with DNS.                  |
| domain-search        | Domain search list used when resolving hostnames with DNS.           |
| domain-name-servers  | List of DNS name servers available to the client.                    |
| host-name            | Name of the client.  |
| nis-domain           | Name of the client's NIS (Sun Network Information Services) domain.  |
| nis-servers          | List of IP addresses indicating NIS servers available to the client. |
| ntp-servers          | List of IP addresses indicating NTP servers available to the client. |
| interface-mtu        | MTU to use on this interface.  |
| netbios-name-servers | List of RFC 1001/1002 NBNS name servers.                             |
| netbios-scope        | NetBIOS over TCP/IP scope parameter for the client.                  |

## Default value

```
subnet-mask
broadcast-address
time-offset
```

```
routers
domain-name
domain-search
domain-name-servers
host-name
nis-domain
nis-servers
ntp-servers
interface-mtu
```

### current-lease (state only)

Current lease.

### fixed-address (state only)

The IPv4 address on the interface.

```
vrouter> show state vrf <vrf> interface physical <physical> ipv4 dhcp current-lease_
↳fixed-address
```

### renew (state only)

Time at which the client should begin trying to contact its server to renew its lease.

```
vrouter> show state vrf <vrf> interface physical <physical> ipv4 dhcp current-lease_
↳renew
```

### rebind (state only)

Time at which the client should begin to try to contact any dhcp server to renew its lease.

```
vrouter> show state vrf <vrf> interface physical <physical> ipv4 dhcp current-lease_
↳rebind
```

## expire (state only)

Time at which the client must stop using a lease if it has not been able to renew it.

```
vrouter> show state vrf <vrf> interface physical <physical> ipv4 dhcp current-lease_
↳ expire
```

## ipv6

Parameters for the IPv6 address family.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv6
```

## enabled

Controls whether IPv6 is enabled or disabled on this interface. When IPv6 is enabled, this interface is connected to an IPv6 stack, and the interface can send and receive IPv6 packets.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv6
vrouter running ipv6# enabled true|false
```

## Default value

true

## address

The list of configured IPv6 addresses on the interface.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv6
vrouter running ipv6# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <X:X::X:X>       | An IPv6 address.  |
| <X:X::X:X/M>     | A masked IPv6 address: address and prefix of that subnet. |

## peer

The IPv6 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv6 address. |
|------|------------------|

## origin (state only)

The origin of this address, e.g., static, dhcp, etc.

```
vrouter> show state vrf <vrf> interface physical <physical> ipv6 address <address> ↵  
↪origin
```

## status (state only)

The status of an address. Most of the states correspond to states from the IPv6 Stateless Address Autoconfiguration protocol.

```
vrouter> show state vrf <vrf> interface physical <physical> ipv6 address <address> ↵  
↪status
```

## neighbor

List of IPv6 neighbors.

```
vrouter running config# vrf <vrf> interface physical <physical> ipv6  
vrouter running ipv6# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv6 address. |
|------------|------------------|

## link-layer-address (mandatory)

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

### router (state only)

Indicates that the neighbor node acts as a router.

```
vrouter> show state vrf <vrf> interface physical <physical> ipv6 neighbor <neighbor>_  
↳router
```

### state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface physical <physical> ipv6 neighbor <neighbor>_  
↳state
```

### network-stack

Network stack parameters for this interface.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack
```

### ipv4

IPv4 parameters.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv4
```

### send-redirects

Send ICMP redirect if host is on the same network than gateway.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv4  
vrouter running ipv4# send-redirects true|false
```

### accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv4
vrouter running ipv4# accept-redirects true|false
```

### accept-source-route

Accept packets with source route option. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv4
vrouter running ipv4# accept-source-route true|false
```

### arp-announce

Define different restriction levels for announcing the local source IP address from IP packets in ARP requests sent on interface. Increasing the restriction level gives more chance for receiving answer from the resolved target while decreasing the level announces more valid sender’s information.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv4
vrouter running ipv4# arp-announce ARP-ANNOUNCE
```

| ARP-ANNOUNCE values | Description  |
|---------------------|--|
| any                 | Use any local address, configured on any interface.  |
| avoid-not-in-subnet | Try to avoid local addresses that are not in the target’s subnet for this interface. This mode is useful when target hosts reachable via this interface require the source IP address in ARP requests to be part of their logical network configured on the receiving interface. When we generate the request we will check all our subnets that include the target IP and will preserve the source address if it is from such subnet. If there is no such subnet we select source address according to the rules for level 2, ‘best-local’.   |
| best-local          | Always use the best local address for this target. In this mode we ignore the source address in the IP packet and try to select local address that we prefer for talks with the target host. Such local address is selected by looking for primary IP addresses on all our subnets on the outgoing interface that include the target IP address. If no suitable local address is found we select the first local address we have on the outgoing interface or on all other interfaces, with the hope we will receive reply for our request and even sometimes no matter the source IP address we announce. |

## arp-filter

Allows to have multiple network interfaces on the same subnet, and have the ARPs for each interface be answered based on whether or not the kernel would route a packet from the ARP'd IP out that interface (therefore you must use source based routing for this to work). In other words it allows control of which cards (usually 1) will respond to an arp request.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv4
vrouter running ipv4# arp-filter true|false
```

## arp-ignore

Define different modes for sending replies in response to received ARP requests that resolve local target IP addresses.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv4
vrouter running ipv4# arp-ignore ARP-IGNORE
```

| ARP-IGNORE values          | Description  |
|----------------------------|--|
| any                        | Reply for any local target IP address, configured on any interface.  |
| check-interface            | Reply only if the target IP address is local address configured on the incoming interface.   |
| check-interface-and-subnet | Reply only if the target IP address is local address configured on the incoming interface and both with the sender's IP address are part from same subnet on this interface. |
| ignore-scope               | Do not reply for local addresses configured with scope host, only resolutions for global and link addresses are replied.   |
| ignore-all                 | Do not reply for all local addresses.  |

## log-invalid-addresses

Log packets with impossible addresses.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv4
vrouter running ipv4# log-invalid-addresses true|false
```

## ipv6

IPv6 parameters.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv6
```

## autoconfiguration

Autoconfigure addresses using Prefix Information in Router Advertisements.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv6
vrouter running ipv6# autoconfiguration true|false
```

## accept-router-advert

Accept Router Advertisements.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv6
vrouter running ipv6# accept-router-advert ACCEPT-ROUTER-ADVERT
```

| ACCEPT-ROUTER-ADVERT values | Description   |
|-----------------------------|---|
| never                       | Do not accept Router Advertisements.                        |
| norouter-mode               | Accept Router Advertisements if forwarding is disabled.     |
| always                      | Accept Router Advertisements even if forwarding is enabled. |

## accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv6
vrouter running ipv6# accept-redirects true|false
```

## accept-source-route

Accept packets with source route option.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv6
vrouter running ipv6# accept-source-route true|false
```

## router-solicitations

Number of Router Solicitations to send until assuming no routers are present.

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv6
vrouter running ipv6# router-solicitations <int16>
```

## use-temporary-addresses

Preference for Privacy Extensions (RFC4941). Not applied to point-to-point and loopback devices (always 0).

```
vrouter running config# vrf <vrf> interface physical <physical> network-stack ipv6
vrouter running ipv6# use-temporary-addresses USE-TEMPORARY-ADDRESSES
```

| USE-TEMPORARY-ADDRESS values | Description   |
|------------------------------|---|
| never                        | Disable Privacy Extensions, i.e. use the public address, subnet prefix/interface id, where interface id is always the same. |
| prefer-public-addresses      | Enable Privacy Extensions, but prefer public addresses over temporary addresses.  |
| always                       | Enable Privacy Extensions and prefer temporary addresses over public addresses.   |

## ethernet

Top-level container for Ethernet configuration.

```
vrouter running config# vrf <vrf> interface physical <physical> ethernet
```

## mac-address

Assigns a MAC address to the Ethernet interface. If not specified, the corresponding operational state leaf is expected to show the system-assigned MAC address.

```
vrouter running config# vrf <vrf> interface physical <physical> ethernet
vrouter running ethernet# mac-address MAC-ADDRESS
```

| MAC-ADDRESS | Description  |
|-------------|--|
| MAC-ADDRESS | An IEEE 802 unicast MAC address i.e. the second digit is an even number. Moreover the mac address must not be 00:00:00:00:00:00. |

## auto-negotiate

Set to true to request the interface to auto-negotiate transmission parameters with its peer interface. When set to false, the transmission parameters must be specified manually.

```
vrouter running config# vrf <vrf> interface physical <physical> ethernet
vrouter running ethernet# auto-negotiate true|false
```

## duplex-mode

Force the duplex mode. If unspecified and auto-negotiate is true, the interface should negotiate the duplex mode directly (typically full- duplex). When auto-negotiate is false, duplex-mode must be specified.

```
vrouter running config# vrf <vrf> interface physical <physical> ethernet
vrouter running ethernet# duplex-mode DUPLEX-MODE
```

| DUPLEX-MODE values | Description       |
|--------------------|-------------------|
| full               | Full duplex mode. |
| half               | Half duplex mode. |

## port-speed

Force the port speed. If unspecified and auto-negotiate is true, the interface should negotiate the port speed directly. When auto- negotiate is false, port-speed must be specified.

```
vrouter running config# vrf <vrf> interface physical <physical> ethernet
vrouter running ethernet# port-speed PORT-SPEED
```

| PORT-SPEED values | Description  |
|-------------------|--|
| 10mb              | 10 Mbps Ethernet.  |
| 100mb             | 100 Mbps Ethernet.   |
| 1gb               | 1 Gbps Ethernet.   |
| 10gb              | 10 Gbps Ethernet.  |
| 25gb              | 25 Gbps Ethernet.  |
| 40gb              | 40 Gbps Ethernet.  |
| 50gb              | 50 Gbps Ethernet.  |
| 100gb             | 100 Gbps Ethernet.   |
| unknown           | Interface speed is unknown. Systems may report unknown when an interface is down or unpopulated (e.g., pluggable not present). |

## flow-control-rx

Enable or disable ingress flow control for this interface. Ethernet flow control is a mechanism by which a receiver may send PAUSE frames to a sender to stop transmission for a specified time. This setting should override auto-negotiated flow control settings. If left unspecified, and auto-negotiate is true, flow control mode is negotiated with the peer interface.

```
vrouter running config# vrf <vrf> interface physical <physical> ethernet  
vrouter running ethernet# flow-control-rx true|false
```

## flow-control-tx

Enable or disable egress flow control for this interface. Ethernet flow control is a mechanism by which a receiver may send PAUSE frames to a sender to stop transmission for a specified time. This setting should override auto-negotiated flow control settings. If left unspecified, and auto-negotiate is true, flow control mode is negotiated with the peer interface.

```
vrouter running config# vrf <vrf> interface physical <physical> ethernet  
vrouter running ethernet# flow-control-tx true|false
```

## qos

---

**Note:** requires a Turbo Router Network License.

---

QoS configuration.

```
vrouter running config# vrf <vrf> interface physical <physical> qos
```

## ingress

Ingress QoS configuration.

```
vrouter running config# vrf <vrf> interface physical <physical> qos ingress
```

## rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface physical <physical> qos ingress rate-limit
```

## policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface physical <physical> qos ingress rate-limit
vrouter running rate-limit# policer <leafref>
```

## shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface physical <physical> qos ingress rate-limit
vrouter running rate-limit# shared-policer <leafref>
```

## policer (state only)

Traffic policer.

## bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface physical <physical> qos ingress rate-limit
↳policer bandwidth
```

## burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface physical <physical> qos ingress rate-limit
↳policer burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface physical <physical> qos ingress rate-limit.  
↳policer excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface physical <physical> qos ingress rate-limit.  
↳policer excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface physical <physical> qos ingress rate-limit.  
↳policer shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos ingress rate-limit.  
↳policer stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos ingress rate-limit  
↳policer stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos ingress rate-limit  
↳policer stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos ingress rate-limit  
↳policer stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface physical <physical> qos ingress rate-limit  
↳policer stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface physical <physical> qos ingress rate-limit  
↳policer stats drop-bytes
```

## egress

Egress QoS configuration.

```
vrouter running config# vrf <vrf> interface physical <physical> qos egress
```

## scheduler (config only)

Scheduler defined in the QoS context.

```
vrouter running config# vrf <vrf> interface physical <physical> qos egress  
vrouter running egress# scheduler <leafref>
```

## rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface physical <physical> qos egress rate-limit
```

## policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface physical <physical> qos egress rate-limit  
vrouter running rate-limit# policer <leafref>
```

## shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface physical <physical> qos egress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

### shaper (config only)

Traffic shaper defined in the QoS context.

```
vrouter running config# vrf <vrf> interface physical <physical> qos egress rate-limit
vrouter running rate-limit# shaper <leafref>
```

### policer (state only)

Traffic policer.

### bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress rate-limit
↳policer bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress rate-limit
↳policer burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress rate-limit
↳policer excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress rate-limit.  
↳policer excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress rate-limit.  
↳policer shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress rate-limit.  
↳policer stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress rate-limit.  
↳policer stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress rate-limit_
↳policer stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress rate-limit_
↳policer stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress rate-limit_
↳policer stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress rate-limit_
↳policer stats drop-bytes
```

### shaper (state only)

Traffic shaper.

### bandwidth (state only)

Maximum bandwidth of shaped traffic.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress rate-limit_
↳shaper bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouters> show state vrf <vrf> interface physical <physical> qos egress rate-limit.  
↳shaper burst
```

### layer1-overhead (state only)

Number of bytes added by the underlying protocol on each packet.

```
vrouters> show state vrf <vrf> interface physical <physical> qos egress rate-limit.  
↳shaper layer1-overhead
```

### queue-size (state only)

Number of packets that can be saved in the delay queue. If a scheduler is also configured on the interface, this value is not used, the queues of the scheduler are used as delay queues. The value is rounded up to the nearest power of 2.

```
vrouters> show state vrf <vrf> interface physical <physical> qos egress rate-limit.  
↳shaper queue-size
```

### stats (state only)

Traffic shaper statistics.

### pass-packets (state only)

Number of packets sent.

```
vrouters> show state vrf <vrf> interface physical <physical> qos egress rate-limit.  
↳shaper stats pass-packets
```

### drop-packets (state only)

Number of packets dropped.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress rate-limit_
↳shaper stats drop-packets
```

### scheduler (state only)

Scheduler state.

### core (state only)

Core used by the scheduler.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler core
```

### pq (state only)

Priority Queueing state.

### nb-queue (state only)

Number of Priority Queueing queues available in the scheduler.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq nb-
↳queue
```

### queue (state only)

List of Priority Queueing queues.

### size (state only)

Size of the queue in packets. The value is rounded up to the nearest power of 2.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> size
```

### policer (state only)

Queue's input policer.

### bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> policer bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> policer burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> policer excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> policer excess-burst
```

### stats (state only)

Queue's input policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> policer stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> policer stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> policer stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> policer stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> policer stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> policer stats drop-bytes
```

### shaper (state only)

Queue's output shaper.

### bandwidth (state only)

Maximum bandwidth of shaped traffic.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> shaper bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> shaper burst
```

### layer1-overhead (state only)

Number of bytes added by the underlying protocol on each packet.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> shaper layer1-overhead
```

### queue-size (state only)

Number of packets that can be saved in the delay queue. If a scheduler is also configured on the interface, this value is not used, the queues of the scheduler are used as delay queues. The value is rounded up to the nearest power of 2.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> shaper queue-size
```

### stats (state only)

Queue's output shaper statistics.

### pass-packets (state only)

Number of packets sent.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> shaper stats pass-packets
```

### drop-packets (state only)

Number of packets dropped.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> shaper stats drop-packets
```

**class (state only)**

Classes assigned to the queue.

**stats (state only)**

Class statistics.

**match-packets (state only)**

Number of packets matched.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> class <string> stats match-packets
```

**stats (state only)**

Queue statistics.

**enqueue-packets (state only)**

Number of packets enqueued.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> stats enqueue-packets
```

**xmit-packets (state only)**

Number of packets sent.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq_
↳queue <uint32> stats xmit-packets
```

### drop-queue-full (state only)

Number of packets dropped.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pq-  
↳queue <uint32> stats drop-queue-full
```

### pb-dwrr (state only)

Priority-Based Deficit Weighted Round Robin description.

### nb-queue (state only)

Number of PB-DWRR queues available in the scheduler.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr nb-queue
```

### queue (state only)

List of PB-DWRR queues.

### size (state only)

Size of the queue in packets. The value is rounded up to the nearest power of 2.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> size
```

### quantum (state only)

Quantum of the queue in bytes. Relevant only if priority is low.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> quantum
```

### priority (state only)

Priority of the queue (low or high).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> priority
```

### policer (state only)

Queue's input policer.

### bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> policer bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> policer burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> policer excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> policer excess-burst
```

### stats (state only)

Queue's input policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> policer stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> policer stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> policer stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> policer stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> policer stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> policer stats drop-bytes
```

### shaper (state only)

Queue's output shaper.

### bandwidth (state only)

Maximum bandwidth of shaped traffic.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> shaper bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> shaper burst
```

### layer1-overhead (state only)

Number of bytes added by the underlying protocol on each packet.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> shaper layer1-overhead
```

### queue-size (state only)

Number of packets that can be saved in the delay queue. If a scheduler is also configured on the interface, this value is not used, the queues of the scheduler are used as delay queues. The value is rounded up to the nearest power of 2.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> shaper queue-size
```

### stats (state only)

Queue's output shaper statistics.

### pass-packets (state only)

Number of packets sent.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> shaper stats pass-packets
```

### drop-packets (state only)

Number of packets dropped.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> shaper stats drop-packets
```

**class (state only)**

Classes assigned to the queue.

**stats (state only)**

Class statistics.

**match-packets (state only)**

Number of packets matched.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> class <string> stats match-packets
```

**stats (state only)**

Queue statistics.

**enqueue-packets (state only)**

Number of packets enqueued.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> stats enqueue-packets
```

**xmit-packets (state only)**

Number of packets sent.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwrr queue <uint32> stats xmit-packets
```

### drop-queue-full (state only)

Number of packets dropped.

```
vrouter> show state vrf <vrf> interface physical <physical> qos egress scheduler pb-  
↳dwr queue <uint32> stats drop-queue-full
```

### counters (state only)

A collection of interface-related statistics objects.

### in-octets (state only)

The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface physical <physical> counters in-octets
```

### in-unicast-pkts (state only)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface physical <physical> counters in-unicast-pkts
```

### in-discards (state only)

The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface physical <physical> counters in-discards
```

### **in-errors (state only)**

For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface physical <physical> counters in-errors
```

### **out-octets (state only)**

The total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface physical <physical> counters out-octets
```

### **out-unicast-pkts (state only)**

The total number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface physical <physical> counters out-unicast-pkts
```

### **out-discards (state only)**

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface physical <physical> counters out-discards
```

## out-errors (state only)

For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface physical <physical> counters out-errors
```

## svti

---

**Note:** requires a Turbo IPsec Application License.

---

The list of SVTI interfaces on the device.

```
vrouter running config# vrf <vrf> interface svti <svti>
```

|        |                    |
|--------|--------------------|
| <svti> | An interface name. |
|--------|--------------------|

## mtu

Set the max transmission unit size in octets.

```
vrouter running config# vrf <vrf> interface svti <svti>  
vrouter running svti <svti># mtu <uint32>
```

## promiscuous

Set promiscuous mode.

```
vrouter running config# vrf <vrf> interface svti <svti>  
vrouter running svti <svti># promiscuous true|false
```

## description

A textual description of the interface.

```
vrouter running config# vrf <vrf> interface svti <svti>  
vrouter running svti <svti># description <string>
```

## enabled

The desired (administrative) state of the interface.

```
vrouter running config# vrf <vrf> interface svti <svti>  
vrouter running svti <svti># enabled true|false
```

## Default value

true

## svti-id (mandatory)

SVTI ID for association with IPsec policies/SA. Must be unique per link-vrf.

```
vrouter running config# vrf <vrf> interface svti <svti>  
vrouter running svti <svti># svti-id <uint32>
```

## link-vrf

The link vrf name.

```
vrouter running config# vrf <vrf> interface svti <svti>  
vrouter running svti <svti># link-vrf <string>
```

## ifindex (state only)

System assigned number for each interface. Corresponds to ifIndex object in SNMP Interface MIB.

```
vrouter> show state vrf <vrf> interface svti <svti> ifindex
```

### admin-status (state only)

The desired state of the interface. In RFC 7223 this leaf has the same read semantics as ifAdminStatus. Here, it reflects the administrative state as set by enabling or disabling the interface.

```
vrouter> show state vrf <vrf> interface svti <svti> admin-status
```

### oper-status (state only)

The current operational state of the interface. This leaf has the same semantics as ifOperStatus.

```
vrouter> show state vrf <vrf> interface svti <svti> oper-status
```

### last-change (state only)

This timestamp indicates the time of the last state change of the interface (e.g., up-to-down transition). This corresponds to the ifLastChange object in the standard interface MIB. The value is the timestamp in nanoseconds relative to the Unix Epoch (Jan 1, 1970 00:00:00 UTC).

```
vrouter> show state vrf <vrf> interface svti <svti> last-change
```

### link-interface (state only)

Link interface.

```
vrouter> show state vrf <vrf> interface svti <svti> link-interface
```

### ethernet

Top-level container for Ethernet configuration.

```
vrouter running config# vrf <vrf> interface svti <svti> ethernet
```

## mac-address

Assigns a MAC address to the Ethernet interface. If not specified, the corresponding operational state leaf is expected to show the system-assigned MAC address.

```
vrouter running config# vrf <vrf> interface svti <svti> ethernet
vrouter running ethernet# mac-address MAC-ADDRESS
```

|             |  |
|-------------|--|
| MAC-ADDRESS | An IEEE 802 unicast MAC address i.e. the second digit is an even number. Moreover the mac address must not be 00:00:00:00:00:00. |
|-------------|--|

## ipv4

Parameters for the IPv4 address family.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4
```

### enabled

Controls whether IPv4 is enabled or disabled on this interface. When IPv4 is enabled, this interface is connected to an IPv4 stack, and the interface can send and receive IPv4 packets.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4
vrouter running ipv4# enabled true|false
```

### Default value

true

### address

The list of configured IPv4 addresses on the interface.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4
vrouter running ipv4# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <A.B.C.D>        | An IPv4 address.  |
| <A.B.C.D/M>      | A masked IPv4 address: address and prefix of that subnet. |

**peer**

The IPv4 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv4 address. |
|------|------------------|

**origin (state only)**

The origin of this address, e.g., statically configured, assigned by DHCP, etc..

```
vrouters> show state vrf <vrf> interface svti <svti> ipv4 address <address> origin
```

**neighbor**

A list of mappings from IPv4 addresses to link-layer addresses. Entries in this list are used as static entries in the ARP Cache.

```
vrouters running config# vrf <vrf> interface svti <svti> ipv4
vrouters running ipv4# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv4 address. |
|------------|------------------|

**link-layer-address (mandatory)**

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

### state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface svti <svti> ipv4 neighbor <neighbor> state
```

### dhcp

DHCP client configuration.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4 dhcp
```

### enabled

Enable or disable DHCP.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4 dhcp  
vrouter running dhcp# enabled true|false
```

### Default value

true

### timeout

Time before deciding that it's not going to be able to contact a server.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4 dhcp  
vrouter running dhcp# timeout <uint32>
```

### Default value

60

### retry

Time before trying again to contact a DHCP server.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4 dhcp  
vrouter running dhcp# retry <uint32>
```

### Default value

300

### select-timeout

Time at which the client stops waiting for other offers from servers.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4 dhcp  
vrouter running dhcp# select-timeout <uint32>
```

#### Default value

0

### reboot

Time after trying to reacquire its old address before trying to discover a new address.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4 dhcp  
vrouter running dhcp# reboot <uint32>
```

#### Default value

10

### initial-interval

Time between the first attempt to reach a server and the second attempt to reach a server.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4 dhcp  
vrouter running dhcp# initial-interval <uint32>
```

#### Default value

10

### dhcp-lease-time

Requested lease time.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4 dhcp  
vrouter running dhcp# dhcp-lease-time <uint32>
```

#### Default value

7200

### dhcp-client-identifier-ascii

DHCP client identifier (ASCII).

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4 dhcp  
vrouter running dhcp# dhcp-client-identifier-ascii <string>
```

### dhcp-client-identifier-hexa

DHCP client identifier (hexadecimal).

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4 dhcp  
vrouter running dhcp# dhcp-client-identifier-hexa <string>
```

### host-name

DHCP client name.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4 dhcp  
vrouter running dhcp# host-name <string>
```

### request

DHCP requests.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv4 dhcp  
vrouter running dhcp# request REQUEST
```

| REQUEST values       | Description  |
|----------------------|--|
| subnet-mask          | Client's subnet mask.  |
| broadcast-address    | Broadcast address in use on the client's subnet.                     |
| time-offset          | Offset of the client's subnet in seconds from UTC.                   |
| routers              | List of IP addresses for routers on the client's subnet.             |
| domain-name          | Domain name used when resolving hostnames with DNS.                  |
| domain-search        | Domain search list used when resolving hostnames with DNS.           |
| domain-name-servers  | List of DNS name servers available to the client.                    |
| host-name            | Name of the client.  |
| nis-domain           | Name of the client's NIS (Sun Network Information Services) domain.  |
| nis-servers          | List of IP addresses indicating NIS servers available to the client. |
| ntp-servers          | List of IP addresses indicating NTP servers available to the client. |
| interface-mtu        | MTU to use on this interface.  |
| netbios-name-servers | List of RFC 1001/1002 NBNS name servers.                             |
| netbios-scope        | NetBIOS over TCP/IP scope parameter for the client.                  |

### Default value

subnet-mask  
 broadcast-address  
 time-offset  
 routers  
 domain-name  
 domain-search  
 domain-name-servers  
 host-name  
 nis-domain  
 nis-servers  
 ntp-servers  
 interface-mtu

### current-lease (state only)

Current lease.

### fixed-address (state only)

The IPv4 address on the interface.

```
vrouter> show state vrf <vrf> interface svti <svti> ipv4 dhcp current-lease fixed-  
↪address
```

### renew (state only)

Time at which the client should begin trying to contact its server to renew its lease.

```
vrouter> show state vrf <vrf> interface svti <svti> ipv4 dhcp current-lease renew
```

### rebind (state only)

Time at which the client should begin to try to contact any dhcp server to renew its lease.

```
vrouter> show state vrf <vrf> interface svti <svti> ipv4 dhcp current-lease rebind
```

### expire (state only)

Time at which the client must stop using a lease if it has not been able to renew it.

```
vrouter> show state vrf <vrf> interface svti <svti> ipv4 dhcp current-lease expire
```

## ipv6

Parameters for the IPv6 address family.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv6
```

### enabled

Controls whether IPv6 is enabled or disabled on this interface. When IPv6 is enabled, this interface is connected to an IPv6 stack, and the interface can send and receive IPv6 packets.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv6  
vrouter running ipv6# enabled true|false
```

### Default value

true

## address

The list of configured IPv6 addresses on the interface.

```
vrouter running config# vrf <vrf> interface svti <svti> ipv6
vrouter running ipv6# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <X:X::X:X>       | An IPv6 address.  |
| <X:X::X:X/M>     | A masked IPv6 address: address and prefix of that subnet. |

## peer

The IPv6 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv6 address. |
|------|------------------|

## origin (state only)

The origin of this address, e.g., static, dhcp, etc.

```
vrouter> show state vrf <vrf> interface svti <svti> ipv6 address <address> origin
```

## status (state only)

The status of an address. Most of the states correspond to states from the IPv6 Stateless Address Autoconfiguration protocol.

```
vrouter> show state vrf <vrf> interface svti <svti> ipv6 address <address> status
```

## neighbor

List of IPv6 neighbors.

```
vrouters running config# vrf <vrf> interface svti <svti> ipv6
vrouters running ipv6# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv6 address. |
|------------|------------------|

### link-layer-address (mandatory)

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

### router (state only)

Indicates that the neighbor node acts as a router.

```
vrouters> show state vrf <vrf> interface svti <svti> ipv6 neighbor <neighbor> router
```

### state (state only)

The state of this neighbor entry.

```
vrouters> show state vrf <vrf> interface svti <svti> ipv6 neighbor <neighbor> state
```

### network-stack

Network stack parameters for this interface.

```
vrouters running config# vrf <vrf> interface svti <svti> network-stack
```

## ipv4

IPv4 parameters.

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv4
```

### send-redirects

Send ICMP redirect if host is on the same network than gateway.

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv4  
vrouter running ipv4# send-redirects true|false
```

### accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv4  
vrouter running ipv4# accept-redirects true|false
```

### accept-source-route

Accept packets with source route option. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv4  
vrouter running ipv4# accept-source-route true|false
```

### arp-announce

Define different restriction levels for announcing the local source IP address from IP packets in ARP requests sent on interface. Increasing the restriction level gives more chance for receiving answer from the resolved target while decreasing the level announces more valid sender's information.

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv4  
vrouter running ipv4# arp-announce ARP-ANNOUNCE
```

| ARP-AND-Description<br>val-ues | Description  |
|--------------------------------|--|
| any                            | Use any local address, configured on any interface.  |
| avoid-not-in-subnet            | Try to avoid local addresses that are not in the target's subnet for this interface. This mode is useful when target hosts reachable via this interface require the source IP address in ARP requests to be part of their logical network configured on the receiving interface. When we generate the request we will check all our subnets that include the target IP and will preserve the source address if it is from such subnet. If there is no such subnet we select source address according to the rules for level 2, 'best-local'.   |
| best-local                     | Always use the best local address for this target. In this mode we ignore the source address in the IP packet and try to select local address that we prefer for talks with the target host. Such local address is selected by looking for primary IP addresses on all our subnets on the outgoing interface that include the target IP address. If no suitable local address is found we select the first local address we have on the outgoing interface or on all other interfaces, with the hope we will receive reply for our request and even sometimes no matter the source IP address we announce. |

### arp-filter

Allows to have multiple network interfaces on the same subnet, and have the ARPs for each interface be answered based on whether or not the kernel would route a packet from the ARP'd IP out that interface (therefore you must use source based routing for this to work). In other words it allows control of which cards (usually 1) will respond to an arp request.

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv4
vrouter running ipv4# arp-filter true|false
```

### arp-ignore

Define different modes for sending replies in response to received ARP requests that resolve local target IP addresses.

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv4
vrouter running ipv4# arp-ignore ARP-IGNORE
```

| ARP-IGNORE values          | Description  |
|----------------------------|--|
| any                        | Reply for any local target IP address, configured on any interface.  |
| check-interface            | Reply only if the target IP address is local address configured on the incoming interface.   |
| check-interface-and-subnet | Reply only if the target IP address is local address configured on the incoming interface and both with the sender's IP address are part from same subnet on this interface. |
| ignore-scope               | Do not reply for local addresses configured with scope host, only resolutions for global and link addresses are replied.   |
| ignore-all                 | Do not reply for all local addresses.  |

### log-invalid-addresses

Log packets with impossible addresses.

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv4
vrouter running ipv4# log-invalid-addresses true|false
```

### ipv6

IPv6 parameters.

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv6
```

### autoconfiguration

Autoconfigure addresses using Prefix Information in Router Advertisements.

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv6
vrouter running ipv6# autoconfiguration true|false
```

### accept-router-advert

Accept Router Advertisements.

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv6
vrouter running ipv6# accept-router-advert ACCEPT-ROUTER-ADVERT
```

| ACCEPT-ROUTER-ADVERT values | Description   |
|-----------------------------|---|
| never                       | Do not accept Router Advertisements.                        |
| norouter-mode               | Accept Router Advertisements if forwarding is disabled.     |
| always                      | Accept Router Advertisements even if forwarding is enabled. |

### accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router.

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv6
vrouter running ipv6# accept-redirects true|false
```

### accept-source-route

Accept packets with source route option.

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv6
vrouter running ipv6# accept-source-route true|false
```

### router-solicitations

Number of Router Solicitations to send until assuming no routers are present.

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv6
vrouter running ipv6# router-solicitations <int16>
```

### use-temporary-addresses

Preference for Privacy Extensions (RFC4941). Not applied to point-to-point and loopback devices (always 0).

```
vrouter running config# vrf <vrf> interface svti <svti> network-stack ipv6
vrouter running ipv6# use-temporary-addresses USE-TEMPORARY-ADDRESSES
```

| USE-TEMPORARY-ADDRESSES values | Description   |
|--------------------------------|---|
| never                          | Disable Privacy Extensions, i.e. use the public address, subnet prefix/interface id, where interface id is always the same. |
| prefer-public-addresses        | Enable Privacy Extensions, but prefer public addresses over temporary addresses.  |
| always                         | Enable Privacy Extensions and prefer temporary addresses over public addresses.   |

## qos

---

**Note:** requires a Turbo Router Network License.

---

QoS configuration.

```
vrouter running config# vrf <vrf> interface svti <svti> qos
```

## ingress

Ingress QoS configuration.

```
vrouter running config# vrf <vrf> interface svti <svti> qos ingress
```

## rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface svti <svti> qos ingress rate-limit
```

## policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface svti <svti> qos ingress rate-limit  
vrouter running rate-limit# policer <leafref>
```

## shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface svti <svti> qos ingress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

### **policer (state only)**

Traffic policer.

### **bandwidth (state only)**

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface svti <svti> qos ingress rate-limit policer.  
↳bandwidth
```

### **burst (state only)**

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface svti <svti> qos ingress rate-limit policer.  
↳burst
```

### **excess-bandwidth (state only)**

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface svti <svti> qos ingress rate-limit policer.  
↳excess-bandwidth
```

### **excess-burst (state only)**

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface svti <svti> qos ingress rate-limit policer.  
↳excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface svti <svti> qos ingress rate-limit policer_
↳shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouter> show state vrf <vrf> interface svti <svti> qos ingress rate-limit policer_
↳stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouter> show state vrf <vrf> interface svti <svti> qos ingress rate-limit policer_
↳stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouter> show state vrf <vrf> interface svti <svti> qos ingress rate-limit policer_
↳stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouter> show state vrf <vrf> interface svti <svti> qos ingress rate-limit policer_
↳stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface svti <svti> qos ingress rate-limit policer.  
↳stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface svti <svti> qos ingress rate-limit policer.  
↳stats drop-bytes
```

### egress

Egress QoS configuration.

```
vrouter running config# vrf <vrf> interface svti <svti> qos egress
```

### rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface svti <svti> qos egress rate-limit
```

### policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface svti <svti> qos egress rate-limit  
vrouter running rate-limit# policer <leafref>
```

### shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface svti <svti> qos egress rate-limit
vrouter running rate-limit# shared-policer <leafref>
```

### policer (state only)

Traffic policer.

### bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface svti <svti> qos egress rate-limit policer
↳bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface svti <svti> qos egress rate-limit policer burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface svti <svti> qos egress rate-limit policer
↳excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface svti <svti> qos egress rate-limit policer.  
↳excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface svti <svti> qos egress rate-limit policer.  
↳shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface svti <svti> qos egress rate-limit policer.  
↳stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface svti <svti> qos egress rate-limit policer.  
↳stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouters> show state vrf <vrf> interface svti <svti> qos egress rate-limit policer.  
↳stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouters> show state vrf <vrf> interface svti <svti> qos egress rate-limit policer.  
↳stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouters> show state vrf <vrf> interface svti <svti> qos egress rate-limit policer.  
↳stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouters> show state vrf <vrf> interface svti <svti> qos egress rate-limit policer.  
↳stats drop-bytes
```

### counters (state only)

A collection of interface-related statistics objects.

### in-octets (state only)

The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouters> show state vrf <vrf> interface svti <svti> counters in-octets
```

### in-unicast-pkts (state only)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface svti <svti> counters in-unicast-pkts
```

### in-discards (state only)

The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface svti <svti> counters in-discards
```

### in-errors (state only)

For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface svti <svti> counters in-errors
```

### out-octets (state only)

The total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface svti <svti> counters out-octets
```

### out-unicast-pkts (state only)

The total number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re- initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface svti <svti> counters out-unicast-pkts
```

### out-discards (state only)

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface svti <svti> counters out-discards
```

### out-errors (state only)

For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface svti <svti> counters out-errors
```

### svti-template (config only)

---

**Note:** requires a Turbo IPsec Application License.

---

The list of dynamic SVTI interface templates on the device.

```
vrouter running config# vrf <vrf> interface svti-template <string>
```

|          |                             |
|----------|-----------------------------|
| <string> | Dynamic SVTI template name. |
|----------|-----------------------------|

**mtu (config only)**

Set the max transmission unit size in octets.

```
vrouter running config# vrf <vrf> interface svti-template <string>
vrouter running svti-template <string># mtu <uint32>
```

**install-routes-to (config only)**

Install routes via the dynamic SVTI.

```
vrouter running config# vrf <vrf> interface svti-template <string>
vrouter running svti-template <string># install-routes-to INSTALL-ROUTES-TO
```

| INSTALL-ROUTES-TO values | Description  |
|--------------------------|--|
| vip-or-remote-ts         | Routes to the remote VIP, if any, else to the remote traffic selector. |
| remote-ts                | Routes to the remote traffic selector.                                 |
| vip                      | Routes to the remote VIP, if any.                                      |
| none                     | Do not add any route.  |

**Default value**

vip-or-remote-ts

**system-loopback (state only)**

The list of system-loopback interfaces on the device.

**mtu (state only)**

Set the max transmission unit size in octets.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> mtu
```

### promiscuous (state only)

Set promiscuous mode.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> promiscuous
```

### description (state only)

A textual description of the interface.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> description
```

### enabled (state only)

The desired (administrative) state of the interface.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> enabled
```

### ifindex (state only)

System assigned number for each interface. Corresponds to ifIndex object in SNMP Interface MIB.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ifindex
```

### admin-status (state only)

The desired state of the interface. In RFC 7223 this leaf has the same read semantics as ifAdminStatus. Here, it reflects the administrative state as set by enabling or disabling the interface.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> admin-status
```

### oper-status (state only)

The current operational state of the interface. This leaf has the same semantics as ifOperStatus.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> oper-status
```

### last-change (state only)

This timestamp indicates the time of the last state change of the interface (e.g., up-to-down transition). This corresponds to the ifLastChange object in the standard interface MIB. The value is the timestamp in nanoseconds relative to the Unix Epoch (Jan 1, 1970 00:00:00 UTC).

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> last-change
```

### ipv4 (state only)

Parameters for the IPv4 address family.

### enabled (state only)

Controls whether IPv4 is enabled or disabled on this interface. When IPv4 is enabled, this interface is connected to an IPv4 stack, and the interface can send and receive IPv4 packets.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 enabled
```

### address (state only)

The list of configured IPv4 addresses on the interface.

### peer (state only)

The IPv4 address of the remote endpoint for point to point interfaces.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 address  
↪<address> peer
```

### origin (state only)

The origin of this address, e.g., statically configured, assigned by DHCP, etc..

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 address  
↪<address> origin
```

### neighbor (state only)

A list of mappings from IPv4 addresses to link-layer addresses. Entries in this list are used as static entries in the ARP Cache.

### link-layer-address (state only)

The link-layer address of the neighbor node.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4_
↳neighbor <neighbor> link-layer-address
```

### state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4_
↳neighbor <neighbor> state
```

### dhcp (state only)

DHCP client configuration.

### enabled (state only)

Enable or disable DHCP.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳enabled
```

### timeout (state only)

Time before deciding that it's not going to be able to contact a server.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳timeout
```

### retry (state only)

Time before trying again to contact a DHCP server.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳retry
```

### select-timeout (state only)

Time at which the client stops waiting for other offers from servers.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳select-timeout
```

### reboot (state only)

Time after trying to reacquire its old address before trying to discover a new address.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳reboot
```

### initial-interval (state only)

Time between the first attempt to reach a server and the second attempt to reach a server.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳initial-interval
```

### dhcp-lease-time (state only)

Requested lease time.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳dhcp-lease-time
```

### dhcp-client-identifier-ascii (state only)

DHCP client identifier (ASCII).

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳dhcp-client-identifier-ascii
```

### dhcp-client-identifier-hexa (state only)

DHCP client identifier (hexadecimal).

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳dhcp-client-identifier-hexa
```

### host-name (state only)

DHCP client name.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳host-name
```

### request (state only)

DHCP requests.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳request
```

### current-lease (state only)

Current lease.

### fixed-address (state only)

The IPv4 address on the interface.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳current-lease fixed-address
```

### renew (state only)

Time at which the client should begin trying to contact its server to renew its lease.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳current-lease renew
```

### rebind (state only)

Time at which the client should begin to try to contact any dhcp server to renew its lease.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳current-lease rebind
```

### expire (state only)

Time at which the client must stop using a lease if it has not been able to renew it.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv4 dhcp_
↳current-lease expire
```

### ipv6 (state only)

Parameters for the IPv6 address family.

### enabled (state only)

Controls whether IPv6 is enabled or disabled on this interface. When IPv6 is enabled, this interface is connected to an IPv6 stack, and the interface can send and receive IPv6 packets.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv6 enabled
```

### address (state only)

The list of configured IPv6 addresses on the interface.

### peer (state only)

The IPv6 address of the remote endpoint for point to point interfaces.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv6 address  
↳<address> peer
```

### origin (state only)

The origin of this address, e.g., static, dhcp, etc.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv6 address  
↳<address> origin
```

### status (state only)

The status of an address. Most of the states correspond to states from the IPv6 Stateless Address Autoconfiguration protocol.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv6 address  
↳<address> status
```

### neighbor (state only)

List of IPv6 neighbors.

### link-layer-address (state only)

The link-layer address of the neighbor node.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv6.  
↳neighbor <neighbor> link-layer-address
```

### router (state only)

Indicates that the neighbor node acts as a router.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv6.  
↳neighbor <neighbor> router
```

### state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> ipv6.  
↳neighbor <neighbor> state
```

### counters (state only)

A collection of interface-related statistics objects.

### in-octets (state only)

The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> counters in-  
↳octets
```

### in-unicast-pkts (state only)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> counters in-  
↳unicast-pkts
```

### in-discards (state only)

The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> counters in-  
↳discards
```

### in-errors (state only)

For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> counters in-  
↳errors
```

### out-octets (state only)

The total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> counters out-  
↳octets
```

### out-unicast-pkts (state only)

The total number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> counters out-  
↳unicast-pkts
```

### out-discards (state only)

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ‘last-clear’.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> counters out-
↳discards
```

### out-errors (state only)

For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ‘last-clear’.

```
vrouter> show state vrf <vrf> interface system-loopback <system-loopback> counters out-
↳errors
```

### veth

The list of veth interfaces on the device.

```
vrouter running config# vrf <vrf> interface veth <veth>
```

|        |                    |
|--------|--------------------|
| <veth> | An interface name. |
|--------|--------------------|

### mtu

Set the max transmission unit size in octets.

```
vrouter running config# vrf <vrf> interface veth <veth>
vrouter running veth <veth># mtu <uint32>
```

## promiscuous

Set promiscuous mode.

```
vrouter running config# vrf <vrf> interface veth <veth>  
vrouter running veth <veth># promiscuous true|false
```

## description

A textual description of the interface.

```
vrouter running config# vrf <vrf> interface veth <veth>  
vrouter running veth <veth># description <string>
```

## enabled

The desired (administrative) state of the interface.

```
vrouter running config# vrf <vrf> interface veth <veth>  
vrouter running veth <veth># enabled true|false
```

## Default value

true

## link-interface (mandatory)

The other endpoint of the Veth pair.

```
vrouter running config# vrf <vrf> interface veth <veth>  
vrouter running veth <veth># link-interface <leafref>
```

## link-vrf (mandatory)

The link vrf name.

```
vrouter running config# vrf <vrf> interface veth <veth>  
vrouter running veth <veth># link-vrf <string>
```

### ifindex (state only)

System assigned number for each interface. Corresponds to ifIndex object in SNMP Interface MIB.

```
vrouters> show state vrf <vrf> interface veth <veth> ifindex
```

### admin-status (state only)

The desired state of the interface. In RFC 7223 this leaf has the same read semantics as ifAdminStatus. Here, it reflects the administrative state as set by enabling or disabling the interface.

```
vrouters> show state vrf <vrf> interface veth <veth> admin-status
```

### oper-status (state only)

The current operational state of the interface. This leaf has the same semantics as ifOperStatus.

```
vrouters> show state vrf <vrf> interface veth <veth> oper-status
```

### last-change (state only)

This timestamp indicates the time of the last state change of the interface (e.g., up-to-down transition). This corresponds to the ifLastChange object in the standard interface MIB. The value is the timestamp in nanoseconds relative to the Unix Epoch (Jan 1, 1970 00:00:00 UTC).

```
vrouters> show state vrf <vrf> interface veth <veth> last-change
```

### ipv4

Parameters for the IPv4 address family.

```
vrouters running config# vrf <vrf> interface veth <veth> ipv4
```

**enabled**

Controls whether IPv4 is enabled or disabled on this interface. When IPv4 is enabled, this interface is connected to an IPv4 stack, and the interface can send and receive IPv4 packets.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4
vrouter running ipv4# enabled true|false
```

**Default value**

true

**address**

The list of configured IPv4 addresses on the interface.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4
vrouter running ipv4# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <A.B.C.D>        | An IPv4 address.  |
| <A.B.C.D/M>      | A masked IPv4 address: address and prefix of that subnet. |

**peer**

The IPv4 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv4 address. |
|------|------------------|

**origin (state only)**

The origin of this address, e.g., statically configured, assigned by DHCP, etc..

```
vrouter> show state vrf <vrf> interface veth <veth> ipv4 address <address> origin
```

## neighbor

A list of mappings from IPv4 addresses to link-layer addresses. Entries in this list are used as static entries in the ARP Cache.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4
vrouter running ipv4# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv4 address. |
|------------|------------------|

### link-layer-address (mandatory)

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

### state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface veth <veth> ipv4 neighbor <neighbor> state
```

## dhcp

DHCP client configuration.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4 dhcp
```

### enabled

Enable or disable DHCP.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4 dhcp
vrouter running dhcp# enabled true|false
```

### Default value

true

## timeout

Time before deciding that it's not going to be able to contact a server.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4 dhcp  
vrouter running dhcp# timeout <uint32>
```

### Default value

60

## retry

Time before trying again to contact a DHCP server.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4 dhcp  
vrouter running dhcp# retry <uint32>
```

### Default value

300

## select-timeout

Time at which the client stops waiting for other offers from servers.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4 dhcp  
vrouter running dhcp# select-timeout <uint32>
```

### Default value

0

## reboot

Time after trying to reacquire its old address before trying to discover a new address.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4 dhcp  
vrouter running dhcp# reboot <uint32>
```

### Default value

10

### initial-interval

Time between the first attempt to reach a server and the second attempt to reach a server.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4 dhcp  
vrouter running dhcp# initial-interval <uint32>
```

#### Default value

10

### dhcp-lease-time

Requested lease time.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4 dhcp  
vrouter running dhcp# dhcp-lease-time <uint32>
```

#### Default value

7200

### dhcp-client-identifier-ascii

DHCP client identifier (ASCII).

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4 dhcp  
vrouter running dhcp# dhcp-client-identifier-ascii <string>
```

### dhcp-client-identifier-hexa

DHCP client identifier (hexadecimal).

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4 dhcp  
vrouter running dhcp# dhcp-client-identifier-hexa <string>
```

## host-name

DHCP client name.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4 dhcp
vrouter running dhcp# host-name <string>
```

## request

DHCP requests.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv4 dhcp
vrouter running dhcp# request REQUEST
```

| REQUEST values       | Description  |
|----------------------|--|
| subnet-mask          | Client's subnet mask.  |
| broadcast-address    | Broadcast address in use on the client's subnet.                     |
| time-offset          | Offset of the client's subnet in seconds from UTC.                   |
| routers              | List of IP addresses for routers on the client's subnet.             |
| domain-name          | Domain name used when resolving hostnames with DNS.                  |
| domain-search        | Domain search list used when resolving hostnames with DNS.           |
| domain-name-servers  | List of DNS name servers available to the client.                    |
| host-name            | Name of the client.  |
| nis-domain           | Name of the client's NIS (Sun Network Information Services) domain.  |
| nis-servers          | List of IP addresses indicating NIS servers available to the client. |
| ntp-servers          | List of IP addresses indicating NTP servers available to the client. |
| interface-mtu        | MTU to use on this interface.  |
| netbios-name-servers | List of RFC 1001/1002 NBNS name servers.                             |
| netbios-scope        | NetBIOS over TCP/IP scope parameter for the client.                  |

## Default value

```
subnet-mask
broadcast-address
time-offset
routers
domain-name
domain-search
domain-name-servers
host-name
nis-domain
nis-servers
ntp-servers
```

## interface-mtu

### current-lease (state only)

Current lease.

### fixed-address (state only)

The IPv4 address on the interface.

```
vrouter> show state vrf <vrf> interface veth <veth> ipv4 dhcp current-lease fixed-  
↪address
```

### renew (state only)

Time at which the client should begin trying to contact its server to renew its lease.

```
vrouter> show state vrf <vrf> interface veth <veth> ipv4 dhcp current-lease renew
```

### rebind (state only)

Time at which the client should begin to try to contact any dhcp server to renew its lease.

```
vrouter> show state vrf <vrf> interface veth <veth> ipv4 dhcp current-lease rebind
```

### expire (state only)

Time at which the client must stop using a lease if it has not been able to renew it.

```
vrouter> show state vrf <vrf> interface veth <veth> ipv4 dhcp current-lease expire
```

## ipv6

Parameters for the IPv6 address family.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv6
```

**enabled**

Controls whether IPv6 is enabled or disabled on this interface. When IPv6 is enabled, this interface is connected to an IPv6 stack, and the interface can send and receive IPv6 packets.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv6
vrouter running ipv6# enabled true|false
```

**Default value**

true

**address**

The list of configured IPv6 addresses on the interface.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv6
vrouter running ipv6# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <X:X::X:X>       | An IPv6 address.  |
| <X:X::X:X/M>     | A masked IPv6 address: address and prefix of that subnet. |

**peer**

The IPv6 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv6 address. |
|------|------------------|

**origin (state only)**

The origin of this address, e.g., static, dhcp, etc.

```
vrouter> show state vrf <vrf> interface veth <veth> ipv6 address <address> origin
```

**status (state only)**

The status of an address. Most of the states correspond to states from the IPv6 Stateless Address Autoconfiguration protocol.

```
vrouter> show state vrf <vrf> interface veth <veth> ipv6 address <address> status
```

**neighbor**

List of IPv6 neighbors.

```
vrouter running config# vrf <vrf> interface veth <veth> ipv6
vrouter running ipv6# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv6 address. |
|------------|------------------|

**link-layer-address (mandatory)**

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

**router (state only)**

Indicates that the neighbor node acts as a router.

```
vrouter> show state vrf <vrf> interface veth <veth> ipv6 neighbor <neighbor> router
```

**state (state only)**

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface veth <veth> ipv6 neighbor <neighbor> state
```

## network-stack

Network stack parameters for this interface.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack
```

## ipv4

IPv4 parameters.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv4
```

## send-redirects

Send ICMP redirect if host is on the same network than gateway.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv4  
vrouter running ipv4# send-redirects true|false
```

## accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv4  
vrouter running ipv4# accept-redirects true|false
```

## accept-source-route

Accept packets with source route option. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv4  
vrouter running ipv4# accept-source-route true|false
```

### arp-announce

Define different restriction levels for announcing the local source IP address from IP packets in ARP requests sent on interface. Increasing the restriction level gives more chance for receiving answer from the resolved target while decreasing the level announces more valid sender’s information.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv4
vrouter running ipv4# arp-announce ARP-ANNOUNCE
```

| ARP-ANNOUNCE values | Description  |
|---------------------|--|
| any                 | Use any local address, configured on any interface.  |
| avoid-not-in-subnet | Try to avoid local addresses that are not in the target’s subnet for this interface. This mode is useful when target hosts reachable via this interface require the source IP address in ARP requests to be part of their logical network configured on the receiving interface. When we generate the request we will check all our subnets that include the target IP and will preserve the source address if it is from such subnet. If there is no such subnet we select source address according to the rules for level 2, ‘best-local’.   |
| best-local          | Always use the best local address for this target. In this mode we ignore the source address in the IP packet and try to select local address that we prefer for talks with the target host. Such local address is selected by looking for primary IP addresses on all our subnets on the outgoing interface that include the target IP address. If no suitable local address is found we select the first local address we have on the outgoing interface or on all other interfaces, with the hope we will receive reply for our request and even sometimes no matter the source IP address we announce. |

### arp-filter

Allows to have multiple network interfaces on the same subnet, and have the ARPs for each interface be answered based on whether or not the kernel would route a packet from the ARP’d IP out that interface (therefore you must use source based routing for this to work). In other words it allows control of which cards (usually 1) will respond to an arp request.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv4
vrouter running ipv4# arp-filter true|false
```

## arp-ignore

Define different modes for sending replies in response to received ARP requests that resolve local target IP addresses.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv4
vrouter running ipv4# arp-ignore ARP-IGNORE
```

| ARP-IGNORE values          | Description  |
|----------------------------|--|
| any                        | Reply for any local target IP address, configured on any interface.  |
| check-interface            | Reply only if the target IP address is local address configured on the incoming interface.   |
| check-interface-and-subnet | Reply only if the target IP address is local address configured on the incoming interface and both with the sender's IP address are part from same subnet on this interface. |
| ignore-scope               | Do not reply for local addresses configured with scope host, only resolutions for global and link addresses are replied.   |
| ignore-all                 | Do not reply for all local addresses.  |

## log-invalid-addresses

Log packets with impossible addresses.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv4
vrouter running ipv4# log-invalid-addresses true|false
```

## ipv6

IPv6 parameters.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv6
```

## autoconfiguration

Autoconfigure addresses using Prefix Information in Router Advertisements.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv6
vrouter running ipv6# autoconfiguration true|false
```

## accept-router-advert

Accept Router Advertisements.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv6
vrouter running ipv6# accept-router-advert ACCEPT-ROUTER-ADVERT
```

| ACCEPT-ROUTER-ADVERT values | Description   |
|-----------------------------|---|
| never                       | Do not accept Router Advertisements.                        |
| norouter-mode               | Accept Router Advertisements if forwarding is disabled.     |
| always                      | Accept Router Advertisements even if forwarding is enabled. |

## accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv6
vrouter running ipv6# accept-redirects true|false
```

## accept-source-route

Accept packets with source route option.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv6
vrouter running ipv6# accept-source-route true|false
```

## router-solicitations

Number of Router Solicitations to send until assuming no routers are present.

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv6
vrouter running ipv6# router-solicitations <int16>
```

## use-temporary-addresses

Preference for Privacy Extensions (RFC4941). Not applied to point-to-point and loopback devices (always 0).

```
vrouter running config# vrf <vrf> interface veth <veth> network-stack ipv6
vrouter running ipv6# use-temporary-addresses USE-TEMPORARY-ADDRESSES
```

| USE-TEMPORARY-ADDRESSES values | Description   |
|--------------------------------|---|
| never                          | Disable Privacy Extensions, i.e. use the public address, subnet prefix/interface id, where interface id is always the same. |
| prefer-public-addresses        | Enable Privacy Extensions, but prefer public addresses over temporary addresses.  |
| always                         | Enable Privacy Extensions and prefer temporary addresses over public addresses.   |

## ethernet

Top-level container for Ethernet configuration.

```
vrouter running config# vrf <vrf> interface veth <veth> ethernet
```

## mac-address

Assigns a MAC address to the Ethernet interface. If not specified, the corresponding operational state leaf is expected to show the system-assigned MAC address.

```
vrouter running config# vrf <vrf> interface veth <veth> ethernet
vrouter running ethernet# mac-address MAC-ADDRESS
```

|             |  |
|-------------|--|
| MAC-ADDRESS | An IEEE 802 unicast MAC address i.e. the second digit is an even number. Moreover the mac address must not be 00:00:00:00:00:00. |
|-------------|--|

## qos

---

**Note:** requires a Turbo Router Network License.

---

QoS configuration.

```
vrouter running config# vrf <vrf> interface veth <veth> qos
```

## ingress

Ingress QoS configuration.

```
vrouter running config# vrf <vrf> interface veth <veth> qos ingress
```

## rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface veth <veth> qos ingress rate-limit
```

## policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface veth <veth> qos ingress rate-limit  
vrouter running rate-limit# policer <leafref>
```

## shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface veth <veth> qos ingress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

## policer (state only)

Traffic policer.

## bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface veth <veth> qos ingress rate-limit policer ↵  
↵bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface veth <veth> qos ingress rate-limit policer.  
↳burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface veth <veth> qos ingress rate-limit policer.  
↳excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface veth <veth> qos ingress rate-limit policer.  
↳excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface veth <veth> qos ingress rate-limit policer.  
↳shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface veth <veth> qos ingress rate-limit policer.  
↳stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface veth <veth> qos ingress rate-limit policer.  
↳stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface veth <veth> qos ingress rate-limit policer.  
↳stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface veth <veth> qos ingress rate-limit policer.  
↳stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface veth <veth> qos ingress rate-limit policer.  
↳stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface veth <veth> qos ingress rate-limit policer.  
↳stats drop-bytes
```

### egress

Egress QoS configuration.

```
vrouter running config# vrf <vrf> interface veth <veth> qos egress
```

### rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface veth <veth> qos egress rate-limit
```

### policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface veth <veth> qos egress rate-limit  
vrouter running rate-limit# policer <leafref>
```

### shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface veth <veth> qos egress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

### **policer (state only)**

Traffic policer.

### **bandwidth (state only)**

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface veth <veth> qos egress rate-limit policer ↵  
↳bandwidth
```

### **burst (state only)**

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface veth <veth> qos egress rate-limit policer burst
```

### **excess-bandwidth (state only)**

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface veth <veth> qos egress rate-limit policer ↵  
↳excess-bandwidth
```

### **excess-burst (state only)**

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface veth <veth> qos egress rate-limit policer ↵  
↳excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouters> show state vrf <vrf> interface veth <veth> qos egress rate-limit policer.  
↳shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouters> show state vrf <vrf> interface veth <veth> qos egress rate-limit policer.  
↳stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouters> show state vrf <vrf> interface veth <veth> qos egress rate-limit policer.  
↳stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouters> show state vrf <vrf> interface veth <veth> qos egress rate-limit policer.  
↳stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouters> show state vrf <vrf> interface veth <veth> qos egress rate-limit policer.  
↳stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface veth <veth> qos egress rate-limit policer.  
↳stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface veth <veth> qos egress rate-limit policer.  
↳stats drop-bytes
```

### counters (state only)

A collection of interface-related statistics objects.

### in-octets (state only)

The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface veth <veth> counters in-octets
```

### in-unicast-pkts (state only)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface veth <veth> counters in-unicast-pkts
```

### in-discards (state only)

The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface veth <veth> counters in-discards
```

### in-errors (state only)

For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface veth <veth> counters in-errors
```

### out-octets (state only)

The total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface veth <veth> counters out-octets
```

### out-unicast-pkts (state only)

The total number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface veth <veth> counters out-unicast-pkts
```

### out-discards (state only)

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface veth <veth> counters out-discards
```

### out-errors (state only)

For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface veth <veth> counters out-errors
```

### vlan

The list of VLAN interfaces on the device.

```
vrouter running config# vrf <vrf> interface vlan <vlan>
```

|        |                    |
|--------|--------------------|
| <vlan> | An interface name. |
|--------|--------------------|

### mtu

Set the max transmission unit size in octets.

```
vrouter running config# vrf <vrf> interface vlan <vlan>  
vrouter running vlan <vlan># mtu <uint32>
```

## promiscuous

Set promiscuous mode.

```
vrouter running config# vrf <vrf> interface vlan <vlan>  
vrouter running vlan <vlan># promiscuous true|false
```

## description

A textual description of the interface.

```
vrouter running config# vrf <vrf> interface vlan <vlan>  
vrouter running vlan <vlan># description <string>
```

## enabled

The desired (administrative) state of the interface.

```
vrouter running config# vrf <vrf> interface vlan <vlan>  
vrouter running vlan <vlan># enabled true|false
```

## Default value

true

## vlan-id (mandatory)

Interface VLAN id.

```
vrouter running config# vrf <vrf> interface vlan <vlan>  
vrouter running vlan <vlan># vlan-id VLAN-ID
```

|         |  |
|---------|--|
| VLAN-ID | Type definition representing a single-tagged VLAN. |
|---------|--|

### link-interface (mandatory)

Create the VLAN on top of this interface.

```
vrouter running config# vrf <vrf> interface vlan <vlan>
vrouter running vlan <vlan># link-interface LINK-INTERFACE
```

|                |                    |
|----------------|--------------------|
| LINK-INTERFACE | An interface name. |
|----------------|--------------------|

### protocol

The VLAN protocol to use.

```
vrouter running config# vrf <vrf> interface vlan <vlan>
vrouter running vlan <vlan># protocol PROTOCOL
```

| PROTOCOL values | Description    |
|-----------------|----------------|
| 802.1q          | VLAN protocol. |
| 802.1ad         | QinQ protocol. |

### Default value

802.1q

### link-vrf

The link vrf name.

```
vrouter running config# vrf <vrf> interface vlan <vlan>
vrouter running vlan <vlan># link-vrf <string>
```

### ifindex (state only)

System assigned number for each interface. Corresponds to ifIndex object in SNMP Interface MIB.

```
vrouter> show state vrf <vrf> interface vlan <vlan> ifindex
```

### admin-status (state only)

The desired state of the interface. In RFC 7223 this leaf has the same read semantics as ifAdminStatus. Here, it reflects the administrative state as set by enabling or disabling the interface.

```
vrouter> show state vrf <vrf> interface vlan <vlan> admin-status
```

### oper-status (state only)

The current operational state of the interface. This leaf has the same semantics as ifOperStatus.

```
vrouter> show state vrf <vrf> interface vlan <vlan> oper-status
```

### last-change (state only)

This timestamp indicates the time of the last state change of the interface (e.g., up-to-down transition). This corresponds to the ifLastChange object in the standard interface MIB. The value is the timestamp in nanoseconds relative to the Unix Epoch (Jan 1, 1970 00:00:00 UTC).

```
vrouter> show state vrf <vrf> interface vlan <vlan> last-change
```

## ethernet

Top-level container for Ethernet configuration.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ethernet
```

### mac-address

Assigns a MAC address to the Ethernet interface. If not specified, the corresponding operational state leaf is expected to show the system-assigned MAC address.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ethernet
vrouter running ethernet# mac-address MAC-ADDRESS
```

|             |  |
|-------------|--|
| MAC-ADDRESS | An IEEE 802 unicast MAC address i.e. the second digit is an even number. Moreover the mac address must not be 00:00:00:00:00:00. |
|-------------|--|

## ipv4

Parameters for the IPv4 address family.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4
```

### enabled

Controls whether IPv4 is enabled or disabled on this interface. When IPv4 is enabled, this interface is connected to an IPv4 stack, and the interface can send and receive IPv4 packets.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4
vrouter running ipv4# enabled true|false
```

### Default value

true

### address

The list of configured IPv4 addresses on the interface.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4
vrouter running ipv4# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <A.B.C.D>        | An IPv4 address.  |
| <A.B.C.D/M>      | A masked IPv4 address: address and prefix of that subnet. |

### peer

The IPv4 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv4 address. |
|------|------------------|

**origin (state only)**

The origin of this address, e.g., statically configured, assigned by DHCP, etc..

```
vrouter> show state vrf <vrf> interface vlan <vlan> ipv4 address <address> origin
```

**neighbor**

A list of mappings from IPv4 addresses to link-layer addresses. Entries in this list are used as static entries in the ARP Cache.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4
vrouter running ipv4# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv4 address. |
|------------|------------------|

**link-layer-address (mandatory)**

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

**state (state only)**

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface vlan <vlan> ipv4 neighbor <neighbor> state
```

**dhcp**

DHCP client configuration.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4 dhcp
```

## enabled

Enable or disable DHCP.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4 dhcp
vrouter running dhcp# enabled true|false
```

### Default value

true

## timeout

Time before deciding that it's not going to be able to contact a server.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4 dhcp
vrouter running dhcp# timeout <uint32>
```

### Default value

60

## retry

Time before trying again to contact a DHCP server.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4 dhcp
vrouter running dhcp# retry <uint32>
```

### Default value

300

## select-timeout

Time at which the client stops waiting for other offers from servers.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4 dhcp
vrouter running dhcp# select-timeout <uint32>
```

### Default value

0

## reboot

Time after trying to reacquire its old address before trying to discover a new address.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4 dhcp  
vrouter running dhcp# reboot <uint32>
```

### Default value

10

## initial-interval

Time between the first attempt to reach a server and the second attempt to reach a server.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4 dhcp  
vrouter running dhcp# initial-interval <uint32>
```

### Default value

10

## dhcp-lease-time

Requested lease time.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4 dhcp  
vrouter running dhcp# dhcp-lease-time <uint32>
```

### Default value

7200

## dhcp-client-identifier-ascii

DHCP client identifier (ASCII).

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4 dhcp  
vrouter running dhcp# dhcp-client-identifier-ascii <string>
```

## dhcp-client-identifier-hexa

DHCP client identifier (hexadecimal).

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4 dhcp
vrouter running dhcp# dhcp-client-identifier-hexa <string>
```

## host-name

DHCP client name.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4 dhcp
vrouter running dhcp# host-name <string>
```

## request

DHCP requests.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv4 dhcp
vrouter running dhcp# request REQUEST
```

| REQUEST values       | Description  |
|----------------------|--|
| subnet-mask          | Client's subnet mask.  |
| broadcast-address    | Broadcast address in use on the client's subnet.                     |
| time-offset          | Offset of the client's subnet in seconds from UTC.                   |
| routers              | List of IP addresses for routers on the client's subnet.             |
| domain-name          | Domain name used when resolving hostnames with DNS.                  |
| domain-search        | Domain search list used when resolving hostnames with DNS.           |
| domain-name-servers  | List of DNS name servers available to the client.                    |
| host-name            | Name of the client.  |
| nis-domain           | Name of the client's NIS (Sun Network Information Services) domain.  |
| nis-servers          | List of IP addresses indicating NIS servers available to the client. |
| ntp-servers          | List of IP addresses indicating NTP servers available to the client. |
| interface-mtu        | MTU to use on this interface.  |
| netbios-name-servers | List of RFC 1001/1002 NBNS name servers.                             |
| netbios-scope        | NetBIOS over TCP/IP scope parameter for the client.                  |

## Default value

```
subnet-mask
broadcast-address
time-offset
```

```
routers
domain-name
domain-search
domain-name-servers
host-name
nis-domain
nis-servers
ntp-servers
interface-mtu
```

### current-lease (state only)

Current lease.

### fixed-address (state only)

The IPv4 address on the interface.

```
vrouter> show state vrf <vrf> interface vlan <vlan> ipv4 dhcp current-lease fixed-
↪address
```

### renew (state only)

Time at which the client should begin trying to contact its server to renew its lease.

```
vrouter> show state vrf <vrf> interface vlan <vlan> ipv4 dhcp current-lease renew
```

### rebind (state only)

Time at which the client should begin to try to contact any dhcp server to renew its lease.

```
vrouter> show state vrf <vrf> interface vlan <vlan> ipv4 dhcp current-lease rebind
```

## expire (state only)

Time at which the client must stop using a lease if it has not been able to renew it.

```
vrouter> show state vrf <vrf> interface vlan <vlan> ipv4 dhcp current-lease expire
```

## ipv6

Parameters for the IPv6 address family.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv6
```

## enabled

Controls whether IPv6 is enabled or disabled on this interface. When IPv6 is enabled, this interface is connected to an IPv6 stack, and the interface can send and receive IPv6 packets.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv6
vrouter running ipv6# enabled true|false
```

## Default value

true

## address

The list of configured IPv6 addresses on the interface.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv6
vrouter running ipv6# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <X:X::X:X>       | An IPv6 address.  |
| <X:X::X:X/M>     | A masked IPv6 address: address and prefix of that subnet. |

**peer**

The IPv6 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv6 address. |
|------|------------------|

**origin (state only)**

The origin of this address, e.g., static, dhcp, etc.

```
vrouter> show state vrf <vrf> interface vlan <vlan> ipv6 address <address> origin
```

**status (state only)**

The status of an address. Most of the states correspond to states from the IPv6 Stateless Address Autoconfiguration protocol.

```
vrouter> show state vrf <vrf> interface vlan <vlan> ipv6 address <address> status
```

**neighbor**

List of IPv6 neighbors.

```
vrouter running config# vrf <vrf> interface vlan <vlan> ipv6
vrouter running ipv6# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv6 address. |
|------------|------------------|

**link-layer-address (mandatory)**

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

### router (state only)

Indicates that the neighbor node acts as a router.

```
vrouter> show state vrf <vrf> interface vlan <vlan> ipv6 neighbor <neighbor> router
```

### state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface vlan <vlan> ipv6 neighbor <neighbor> state
```

### network-stack

Network stack parameters for this interface.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack
```

### ipv4

IPv4 parameters.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv4
```

### send-redirects

Send ICMP redirect if host is on the same network than gateway.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv4  
vrouter running ipv4# send-redirects true|false
```

### accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv4  
vrouter running ipv4# accept-redirects true|false
```

### accept-source-route

Accept packets with source route option. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv4
vrouter running ipv4# accept-source-route true|false
```

### arp-announce

Define different restriction levels for announcing the local source IP address from IP packets in ARP requests sent on interface. Increasing the restriction level gives more chance for receiving answer from the resolved target while decreasing the level announces more valid sender’s information.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv4
vrouter running ipv4# arp-announce ARP-ANNOUNCE
```

| ARP-ANNOUNCE values | Description  |
|---------------------|--|
| any                 | Use any local address, configured on any interface.  |
| avoid-not-in-subnet | Try to avoid local addresses that are not in the target’s subnet for this interface. This mode is useful when target hosts reachable via this interface require the source IP address in ARP requests to be part of their logical network configured on the receiving interface. When we generate the request we will check all our subnets that include the target IP and will preserve the source address if it is from such subnet. If there is no such subnet we select source address according to the rules for level 2, ‘best-local’.   |
| best-local          | Always use the best local address for this target. In this mode we ignore the source address in the IP packet and try to select local address that we prefer for talks with the target host. Such local address is selected by looking for primary IP addresses on all our subnets on the outgoing interface that include the target IP address. If no suitable local address is found we select the first local address we have on the outgoing interface or on all other interfaces, with the hope we will receive reply for our request and even sometimes no matter the source IP address we announce. |

### arp-filter

Allows to have multiple network interfaces on the same subnet, and have the ARPs for each interface be answered based on whether or not the kernel would route a packet from the ARP’d IP out that interface (therefore you must use source based routing for this to work). In other words it allows control of which cards (usually 1) will respond to an arp request.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv4
vrouter running ipv4# arp-filter true|false
```

## arp-ignore

Define different modes for sending replies in response to received ARP requests that resolve local target IP addresses.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv4
vrouter running ipv4# arp-ignore ARP-IGNORE
```

| ARP-IGNORE values          | Description  |
|----------------------------|--|
| any                        | Reply for any local target IP address, configured on any interface.  |
| check-interface            | Reply only if the target IP address is local address configured on the incoming interface.   |
| check-interface-and-subnet | Reply only if the target IP address is local address configured on the incoming interface and both with the sender's IP address are part from same subnet on this interface. |
| ignore-scope               | Do not reply for local addresses configured with scope host, only resolutions for global and link addresses are replied.   |
| ignore-all                 | Do not reply for all local addresses.  |

## log-invalid-addresses

Log packets with impossible addresses.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv4
vrouter running ipv4# log-invalid-addresses true|false
```

## ipv6

IPv6 parameters.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv6
```

## autoconfiguration

Autoconfigure addresses using Prefix Information in Router Advertisements.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv6
vrouter running ipv6# autoconfiguration true|false
```

## accept-router-advert

Accept Router Advertisements.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv6
vrouter running ipv6# accept-router-advert ACCEPT-ROUTER-ADVERT
```

| ACCEPT-ROUTER-ADVERT values | Description   |
|-----------------------------|---|
| never                       | Do not accept Router Advertisements.                        |
| norouter-mode               | Accept Router Advertisements if forwarding is disabled.     |
| always                      | Accept Router Advertisements even if forwarding is enabled. |

## accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv6
vrouter running ipv6# accept-redirects true|false
```

## accept-source-route

Accept packets with source route option.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv6
vrouter running ipv6# accept-source-route true|false
```

## router-solicitations

Number of Router Solicitations to send until assuming no routers are present.

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv6
vrouter running ipv6# router-solicitations <int16>
```

## use-temporary-addresses

Preference for Privacy Extensions (RFC4941). Not applied to point-to-point and loopback devices (always 0).

```
vrouter running config# vrf <vrf> interface vlan <vlan> network-stack ipv6
vrouter running ipv6# use-temporary-addresses USE-TEMPORARY-ADDRESSES
```

| USE-TEMPORARY-ADDRESSES values | Description   |
|--------------------------------|---|
| never                          | Disable Privacy Extensions, i.e. use the public address, subnet prefix/interface id, where interface id is always the same. |
| prefer-public-addresses        | Enable Privacy Extensions, but prefer public addresses over temporary addresses.  |
| always                         | Enable Privacy Extensions and prefer temporary addresses over public addresses.   |

## qos

---

**Note:** requires a Turbo Router Network License.

---

QoS configuration.

```
vrouter running config# vrf <vrf> interface vlan <vlan> qos
```

## ingress

Ingress QoS configuration.

```
vrouter running config# vrf <vrf> interface vlan <vlan> qos ingress
```

## rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface vlan <vlan> qos ingress rate-limit
```

### policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface vlan <vlan> qos ingress rate-limit
vrouter running rate-limit# policer <leafref>
```

### shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface vlan <vlan> qos ingress rate-limit
vrouter running rate-limit# shared-policer <leafref>
```

### policer (state only)

Traffic policer.

### bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos ingress rate-limit policer_
↳bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos ingress rate-limit policer_
↳burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos ingress rate-limit policer.  
↳excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos ingress rate-limit policer.  
↳excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos ingress rate-limit policer.  
↳shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos ingress rate-limit policer.  
↳stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos ingress rate-limit policer.  
↳stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos ingress rate-limit policer.  
↳stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos ingress rate-limit policer.  
↳stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos ingress rate-limit policer.  
↳stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos ingress rate-limit policer.  
↳stats drop-bytes
```

## egress

Egress QoS configuration.

```
vrouter running config# vrf <vrf> interface vlan <vlan> qos egress
```

## rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface vlan <vlan> qos egress rate-limit
```

## policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface vlan <vlan> qos egress rate-limit  
vrouter running rate-limit# policer <leafref>
```

## shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface vlan <vlan> qos egress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

## policer (state only)

Traffic policer.

## bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos egress rate-limit policer ↵  
↵bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos egress rate-limit policer burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos egress rate-limit policer ↵  
↵excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos egress rate-limit policer ↵  
↵excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos egress rate-limit policer ↵  
↵shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos egress rate-limit policer.  
↳stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos egress rate-limit policer.  
↳stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos egress rate-limit policer.  
↳stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos egress rate-limit policer.  
↳stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos egress rate-limit policer.  
↳stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface vlan <vlan> qos egress rate-limit policer.  
↳stats drop-bytes
```

### counters (state only)

A collection of interface-related statistics objects.

### in-octets (state only)

The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vlan <vlan> counters in-octets
```

### in-unicast-pkts (state only)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vlan <vlan> counters in-unicast-pkts
```

### in-discards (state only)

The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vlan <vlan> counters in-discards
```

### in-errors (state only)

For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface vlan <vlan> counters in-errors
```

### out-octets (state only)

The total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface vlan <vlan> counters out-octets
```

### out-unicast-pkts (state only)

The total number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface vlan <vlan> counters out-unicast-pkts
```

### out-discards (state only)

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrrouter> show state vrf <vrf> interface vlan <vlan> counters out-discards
```

## out-errors (state only)

For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vlan <vlan> counters out-errors
```

## vxlan

---

**Note:** requires a Turbo Router Network License.

---

The list of VxLAN interfaces on the device.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>
```

|         |                    |
|---------|--------------------|
| <vxlan> | An interface name. |
|---------|--------------------|

## mtu

Set the max transmission unit size in octets.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>  
vrouter running vxlan <vxlan># mtu <uint32>
```

## promiscuous

Set promiscuous mode.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>  
vrouter running vxlan <vxlan># promiscuous true|false
```

## description

A textual description of the interface.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>
vrouter running vxlan <vxlan># description <string>
```

## enabled

The desired (administrative) state of the interface.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>
vrouter running vxlan <vxlan># enabled true|false
```

## Default value

true

## vni (mandatory)

Interface VXLAN Network ID. This ID must be unique.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>
vrouter running vxlan <vxlan># vni VNI
```

|     |   |
|-----|---|
| VNI | Type definition representing VXLAN Segment ID / VXLAN Network Identifier value. |
|-----|---|

## group

The group multicast IP address.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>
vrouter running vxlan <vxlan># group GROUP
```

| GROUP values | Description   |
|--------------|---|
| <A.B.C.D>    | An IPv4 multicast group address, which is in the range of 224.0.0.0 to 239.255.255.255. |
| <X:X::X:X>   | An IPv6 multicast group address, which is in the range of ff00::/8.                     |

## local

The source address that should be used for the Vxlan tunnel. If none is specified an address of the link interface will be used.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>
vrouter running vxlan <vxlan># local LOCAL
```

| LOCAL values | Description      |
|--------------|------------------|
| <A.B.C.D>    | An IPv4 address. |
| <X:X::X:X>   | An IPv6 address. |

## ttl

The time-to-live (or hop limit) that should be utilised for the IP packets used for the tunnel transport.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>
vrouter running vxlan <vxlan># ttl <uint8>
```

## tos

Set the DSCP bits in the Type of Service field.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>
vrouter running vxlan <vxlan># tos <uint8>
```

## link-interface

Route tunneled packets through this interface.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>
vrouter running vxlan <vxlan># link-interface LINK-INTERFACE
```

|                |                    |
|----------------|--------------------|
| LINK-INTERFACE | An interface name. |
|----------------|--------------------|

## link-vrf

The link vrf name.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>  
vrouter running vxlan <vxlan># link-vrf <string>
```

## learning

Enable the registration of unknown source link layer addresses and IP addresses into the VxLAN forwarding database.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>  
vrouter running vxlan <vxlan># learning true|false
```

### Default value

true

## gbp

Enable the Group Policy extension.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>  
vrouter running vxlan <vxlan># gbp true|false
```

### Default value

false

## dst

UDP destination port.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan>  
vrouter running vxlan <vxlan># dst <uint16>
```

### Default value

4789

### ifindex (state only)

System assigned number for each interface. Corresponds to ifIndex object in SNMP Interface MIB.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> ifindex
```

### admin-status (state only)

The desired state of the interface. In RFC 7223 this leaf has the same read semantics as ifAdminStatus. Here, it reflects the administrative state as set by enabling or disabling the interface.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> admin-status
```

### oper-status (state only)

The current operational state of the interface. This leaf has the same semantics as ifOperStatus.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> oper-status
```

### last-change (state only)

This timestamp indicates the time of the last state change of the interface (e.g., up-to-down transition). This corresponds to the ifLastChange object in the standard interface MIB. The value is the timestamp in nanoseconds relative to the Unix Epoch (Jan 1, 1970 00:00:00 UTC).

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> last-change
```

### ethernet

Top-level container for Ethernet configuration.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ethernet
```

## mac-address

Assigns a MAC address to the Ethernet interface. If not specified, the corresponding operational state leaf is expected to show the system-assigned MAC address.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ethernet
vrouter running ethernet# mac-address MAC-ADDRESS
```

|             |  |
|-------------|--|
| MAC-ADDRESS | An IEEE 802 unicast MAC address i.e. the second digit is an even number. Moreover the mac address must not be 00:00:00:00:00:00. |
|-------------|--|

## ipv4

Parameters for the IPv4 address family.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4
```

## enabled

Controls whether IPv4 is enabled or disabled on this interface. When IPv4 is enabled, this interface is connected to an IPv4 stack, and the interface can send and receive IPv4 packets.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4
vrouter running ipv4# enabled true|false
```

## Default value

true

## address

The list of configured IPv4 addresses on the interface.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4
vrouter running ipv4# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <A.B.C.D>        | An IPv4 address.  |
| <A.B.C.D/M>      | A masked IPv4 address: address and prefix of that subnet. |

**peer**

The IPv4 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv4 address. |
|------|------------------|

**origin (state only)**

The origin of this address, e.g., statically configured, assigned by DHCP, etc..

```
vrouters> show state vrf <vrf> interface vxlan <vxlan> ipv4 address <address> origin
```

**neighbor**

A list of mappings from IPv4 addresses to link-layer addresses. Entries in this list are used as static entries in the ARP Cache.

```
vrouters running config# vrf <vrf> interface vxlan <vxlan> ipv4
vrouters running ipv4# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv4 address. |
|------------|------------------|

**link-layer-address (mandatory)**

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

### state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> ipv4 neighbor <neighbor> state
```

### dhcp

DHCP client configuration.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4 dhcp
```

### enabled

Enable or disable DHCP.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4 dhcp  
vrouter running dhcp# enabled true|false
```

### Default value

true

### timeout

Time before deciding that it's not going to be able to contact a server.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4 dhcp  
vrouter running dhcp# timeout <uint32>
```

### Default value

60

### retry

Time before trying again to contact a DHCP server.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4 dhcp  
vrouter running dhcp# retry <uint32>
```

### Default value

300

### select-timeout

Time at which the client stops waiting for other offers from servers.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4 dhcp  
vrouter running dhcp# select-timeout <uint32>
```

#### Default value

0

### reboot

Time after trying to reacquire its old address before trying to discover a new address.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4 dhcp  
vrouter running dhcp# reboot <uint32>
```

#### Default value

10

### initial-interval

Time between the first attempt to reach a server and the second attempt to reach a server.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4 dhcp  
vrouter running dhcp# initial-interval <uint32>
```

#### Default value

10

### dhcp-lease-time

Requested lease time.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4 dhcp  
vrouter running dhcp# dhcp-lease-time <uint32>
```

#### Default value

7200

### dhcp-client-identifier-ascii

DHCP client identifier (ASCII).

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4 dhcp
vrouter running dhcp# dhcp-client-identifier-ascii <string>
```

### dhcp-client-identifier-hexa

DHCP client identifier (hexadecimal).

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4 dhcp
vrouter running dhcp# dhcp-client-identifier-hexa <string>
```

### host-name

DHCP client name.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4 dhcp
vrouter running dhcp# host-name <string>
```

### request

DHCP requests.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv4 dhcp
vrouter running dhcp# request REQUEST
```

| REQUEST values       | Description  |
|----------------------|--|
| subnet-mask          | Client's subnet mask.  |
| broadcast-address    | Broadcast address in use on the client's subnet.                     |
| time-offset          | Offset of the client's subnet in seconds from UTC.                   |
| routers              | List of IP addresses for routers on the client's subnet.             |
| domain-name          | Domain name used when resolving hostnames with DNS.                  |
| domain-search        | Domain search list used when resolving hostnames with DNS.           |
| domain-name-servers  | List of DNS name servers available to the client.                    |
| host-name            | Name of the client.  |
| nis-domain           | Name of the client's NIS (Sun Network Information Services) domain.  |
| nis-servers          | List of IP addresses indicating NIS servers available to the client. |
| ntp-servers          | List of IP addresses indicating NTP servers available to the client. |
| interface-mtu        | MTU to use on this interface.  |
| netbios-name-servers | List of RFC 1001/1002 NBNS name servers.                             |
| netbios-scope        | NetBIOS over TCP/IP scope parameter for the client.                  |

### Default value

subnet-mask  
 broadcast-address  
 time-offset  
 routers  
 domain-name  
 domain-search  
 domain-name-servers  
 host-name  
 nis-domain  
 nis-servers  
 ntp-servers  
 interface-mtu

### current-lease (state only)

Current lease.

**fixed-address (state only)**

The IPv4 address on the interface.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> ipv4 dhcp current-lease fixed-
↪address
```

**renew (state only)**

Time at which the client should begin trying to contact its server to renew its lease.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> ipv4 dhcp current-lease renew
```

**rebind (state only)**

Time at which the client should begin to try to contact any dhcp server to renew its lease.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> ipv4 dhcp current-lease rebind
```

**expire (state only)**

Time at which the client must stop using a lease if it has not been able to renew it.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> ipv4 dhcp current-lease expire
```

**ipv6**

Parameters for the IPv6 address family.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv6
```

**enabled**

Controls whether IPv6 is enabled or disabled on this interface. When IPv6 is enabled, this interface is connected to an IPv6 stack, and the interface can send and receive IPv6 packets.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv6
vrouter running ipv6# enabled true|false
```

**Default value**

true

## address

The list of configured IPv6 addresses on the interface.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> ipv6
vrouter running ipv6# address <address> peer PEER
```

| <address> values | Description   |
|------------------|---|
| <X:X::X:X>       | An IPv6 address.  |
| <X:X::X:X/M>     | A masked IPv6 address: address and prefix of that subnet. |

## peer

The IPv6 address of the remote endpoint for point to point interfaces.

```
peer PEER
```

|      |                  |
|------|------------------|
| PEER | An IPv6 address. |
|------|------------------|

## origin (state only)

The origin of this address, e.g., static, dhcp, etc.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> ipv6 address <address> origin
```

## status (state only)

The status of an address. Most of the states correspond to states from the IPv6 Stateless Address Autoconfiguration protocol.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> ipv6 address <address> status
```

## neighbor

List of IPv6 neighbors.

```
vrouters running config# vrf <vrf> interface vxlan <vxlan> ipv6
vrouters running ipv6# neighbor <neighbor> link-layer-address LINK-LAYER-ADDRESS
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv6 address. |
|------------|------------------|

### link-layer-address (mandatory)

The link-layer address of the neighbor node.

```
link-layer-address LINK-LAYER-ADDRESS
```

|                    |                          |
|--------------------|--------------------------|
| LINK-LAYER-ADDRESS | An IEEE 802 MAC address. |
|--------------------|--------------------------|

### router (state only)

Indicates that the neighbor node acts as a router.

```
vrouters> show state vrf <vrf> interface vxlan <vxlan> ipv6 neighbor <neighbor> router
```

### state (state only)

The state of this neighbor entry.

```
vrouters> show state vrf <vrf> interface vxlan <vxlan> ipv6 neighbor <neighbor> state
```

### network-stack

Network stack parameters for this interface.

```
vrouters running config# vrf <vrf> interface vxlan <vxlan> network-stack
```

## ipv4

IPv4 parameters.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv4
```

### send-redirects

Send ICMP redirect if host is on the same network than gateway.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv4  
vrouter running ipv4# send-redirects true|false
```

### accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv4  
vrouter running ipv4# accept-redirects true|false
```

### accept-source-route

Accept packets with source route option. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv4  
vrouter running ipv4# accept-source-route true|false
```

### arp-announce

Define different restriction levels for announcing the local source IP address from IP packets in ARP requests sent on interface. Increasing the restriction level gives more chance for receiving answer from the resolved target while decreasing the level announces more valid sender's information.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv4  
vrouter running ipv4# arp-announce ARP-ANNOUNCE
```

| ARP-AND-Description<br>val-ues | Description  |
|--------------------------------|--|
| any                            | Use any local address, configured on any interface.  |
| avoid-not-in-subnet            | Try to avoid local addresses that are not in the target's subnet for this interface. This mode is useful when target hosts reachable via this interface require the source IP address in ARP requests to be part of their logical network configured on the receiving interface. When we generate the request we will check all our subnets that include the target IP and will preserve the source address if it is from such subnet. If there is no such subnet we select source address according to the rules for level 2, 'best-local'.   |
| best-local                     | Always use the best local address for this target. In this mode we ignore the source address in the IP packet and try to select local address that we prefer for talks with the target host. Such local address is selected by looking for primary IP addresses on all our subnets on the outgoing interface that include the target IP address. If no suitable local address is found we select the first local address we have on the outgoing interface or on all other interfaces, with the hope we will receive reply for our request and even sometimes no matter the source IP address we announce. |

### arp-filter

Allows to have multiple network interfaces on the same subnet, and have the ARPs for each interface be answered based on whether or not the kernel would route a packet from the ARP'd IP out that interface (therefore you must use source based routing for this to work). In other words it allows control of which cards (usually 1) will respond to an arp request.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv4
vrouter running ipv4# arp-filter true|false
```

### arp-ignore

Define different modes for sending replies in response to received ARP requests that resolve local target IP addresses.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv4
vrouter running ipv4# arp-ignore ARP-IGNORE
```

| ARP-IGNORE values          | Description  |
|----------------------------|--|
| any                        | Reply for any local target IP address, configured on any interface.  |
| check-interface            | Reply only if the target IP address is local address configured on the incoming interface.   |
| check-interface-and-subnet | Reply only if the target IP address is local address configured on the incoming interface and both with the sender's IP address are part from same subnet on this interface. |
| ignore-scope               | Do not reply for local addresses configured with scope host, only resolutions for global and link addresses are replied.   |
| ignore-all                 | Do not reply for all local addresses.  |

### log-invalid-addresses

Log packets with impossible addresses.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv4
vrouter running ipv4# log-invalid-addresses true|false
```

### ipv6

IPv6 parameters.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv6
```

### autoconfiguration

Autoconfigure addresses using Prefix Information in Router Advertisements.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv6
vrouter running ipv6# autoconfiguration true|false
```

### accept-router-advert

Accept Router Advertisements.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv6
vrouter running ipv6# accept-router-advert ACCEPT-ROUTER-ADVERT
```

| ACCEPT-ROUTER-ADVERT values | Description   |
|-----------------------------|---|
| never                       | Do not accept Router Advertisements.                        |
| norouter-mode               | Accept Router Advertisements if forwarding is disabled.     |
| always                      | Accept Router Advertisements even if forwarding is enabled. |

### accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv6
vrouter running ipv6# accept-redirects true|false
```

### accept-source-route

Accept packets with source route option.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv6
vrouter running ipv6# accept-source-route true|false
```

### router-solicitations

Number of Router Solicitations to send until assuming no routers are present.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv6
vrouter running ipv6# router-solicitations <int16>
```

### use-temporary-addresses

Preference for Privacy Extensions (RFC4941). Not applied to point-to-point and loopback devices (always 0).

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> network-stack ipv6
vrouter running ipv6# use-temporary-addresses USE-TEMPORARY-ADDRESSES
```

| USE-TEMPORARY-ADDRESSES values | Description   |
|--------------------------------|---|
| never                          | Disable Privacy Extensions, i.e. use the public address, subnet prefix/interface id, where interface id is always the same. |
| prefer-public-addresses        | Enable Privacy Extensions, but prefer public addresses over temporary addresses.  |
| always                         | Enable Privacy Extensions and prefer temporary addresses over public addresses.   |

## src-range

Range of UDP source ports.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> src-range
```

### <uint16>

Minimal value of source port range.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> src-range  
vrouter running src-range# <uint16>
```

### Default value

49152

### <uint16>

Maximal value of source port range.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> src-range  
vrouter running src-range# <uint16>
```

### Default value

65535

## qos

---

**Note:** requires a Turbo Router Network License.

---

QoS configuration.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> qos
```

## ingress

Ingress QoS configuration.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> qos ingress
```

## rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> qos ingress rate-limit
```

## policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> qos ingress rate-limit  
vrouter running rate-limit# policer <leafref>
```

## shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> qos ingress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

## policer (state only)

Traffic policer.

## bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos ingress rate-limit policer ↵  
↵bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos ingress rate-limit policer.  
↳burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos ingress rate-limit policer.  
↳excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos ingress rate-limit policer.  
↳excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos ingress rate-limit policer.  
↳shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos ingress rate-limit policer_
↳stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos ingress rate-limit policer_
↳stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos ingress rate-limit policer_
↳stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos ingress rate-limit policer_
↳stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos ingress rate-limit policer_
↳stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos ingress rate-limit policer.  
↳stats drop-bytes
```

### egress

Egress QoS configuration.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> qos egress
```

### rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> qos egress rate-limit
```

### policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> qos egress rate-limit  
vrouter running rate-limit# policer <leafref>
```

### shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> qos egress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

### **policer (state only)**

Traffic policer.

### **bandwidth (state only)**

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos egress rate-limit policer_
↳bandwidth
```

### **burst (state only)**

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos egress rate-limit policer_
↳burst
```

### **excess-bandwidth (state only)**

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos egress rate-limit policer_
↳excess-bandwidth
```

### **excess-burst (state only)**

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos egress rate-limit policer_
↳excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos egress rate-limit policer_
↳shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos egress rate-limit policer_
↳stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos egress rate-limit policer_
↳stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos egress rate-limit policer_
↳stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos egress rate-limit policer_
↳stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos egress rate-limit policer.  
↳stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> qos egress rate-limit policer.  
↳stats drop-bytes
```

### counters (state only)

A collection of interface-related statistics objects.

### in-octets (state only)

The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> counters in-octets
```

### in-unicast-pkts (state only)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> counters in-unicast-pkts
```

### in-discards (state only)

The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> counters in-discards
```

### in-errors (state only)

For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> counters in-errors
```

### out-octets (state only)

The total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> counters out-octets
```

### out-unicast-pkts (state only)

The total number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> counters out-unicast-pkts
```

### out-discards (state only)

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> counters out-discards
```

### out-errors (state only)

For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> counters out-errors
```

### xvrf

---

**Note:** requires a Turbo Router Network License.

---

The list of xvrf interfaces on the device.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf>
```

|        |                    |
|--------|--------------------|
| <xvrf> | An interface name. |
|--------|--------------------|

### mtu

Set the max transmission unit size in octets.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf>  
vrouter running xvrf <xvrf># mtu <uint32>
```

## promiscuous

Set promiscuous mode.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf>  
vrouter running xvrf <xvrf># promiscuous true|false
```

## description

A textual description of the interface.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf>  
vrouter running xvrf <xvrf># description <string>
```

## enabled

The desired (administrative) state of the interface.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf>  
vrouter running xvrf <xvrf># enabled true|false
```

## Default value

true

## link-interface (mandatory)

The other endpoint of the xvrf pair.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf>  
vrouter running xvrf <xvrf># link-interface <leafref>
```

## link-vrf (mandatory)

The link vrf name.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf>  
vrouter running xvrf <xvrf># link-vrf <string>
```

### ifindex (state only)

System assigned number for each interface. Corresponds to ifIndex object in SNMP Interface MIB.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> ifindex
```

### admin-status (state only)

The desired state of the interface. In RFC 7223 this leaf has the same read semantics as ifAdminStatus. Here, it reflects the administrative state as set by enabling or disabling the interface.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> admin-status
```

### oper-status (state only)

The current operational state of the interface. This leaf has the same semantics as ifOperStatus.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> oper-status
```

### last-change (state only)

This timestamp indicates the time of the last state change of the interface (e.g., up-to-down transition). This corresponds to the ifLastChange object in the standard interface MIB. The value is the timestamp in nanoseconds relative to the Unix Epoch (Jan 1, 1970 00:00:00 UTC).

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> last-change
```

### qos

---

**Note:** requires a Turbo Router Network License.

---

QoS configuration.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf> qos
```

## ingress

Ingress QoS configuration.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf> qos ingress
```

## rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf> qos ingress rate-limit
```

## policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf> qos ingress rate-limit  
vrouter running rate-limit# policer <leafref>
```

## shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf> qos ingress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

## policer (state only)

Traffic policer.

## bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos ingress rate-limit policer ↵  
↵bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos ingress rate-limit policer.  
↳burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos ingress rate-limit policer.  
↳excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos ingress rate-limit policer.  
↳excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos ingress rate-limit policer.  
↳shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos ingress rate-limit policer.  
↳stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos ingress rate-limit policer.  
↳stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos ingress rate-limit policer.  
↳stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos ingress rate-limit policer.  
↳stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos ingress rate-limit policer.  
↳stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos ingress rate-limit policer.  
↳stats drop-bytes
```

### egress

Egress QoS configuration.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf> qos egress
```

### rate-limit

Rate limit configuration.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf> qos egress rate-limit
```

### policer (config only)

Traffic policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf> qos egress rate-limit  
vrouter running rate-limit# policer <leafref>
```

### shared-policer (config only)

Traffic shared policer defined in the QoS context.

```
vrouter running config# vrf <vrf> interface xvrf <xvrf> qos egress rate-limit  
vrouter running rate-limit# shared-policer <leafref>
```

### **policer (state only)**

Traffic policer.

### **bandwidth (state only)**

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos egress rate-limit policer ↵  
↵-bandwidth
```

### **burst (state only)**

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos egress rate-limit policer burst
```

### **excess-bandwidth (state only)**

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos egress rate-limit policer ↵  
↵-excess-bandwidth
```

### **excess-burst (state only)**

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos egress rate-limit policer ↵  
↵-excess-burst
```

### shared-policer (state only)

Shared policer name.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos egress rate-limit policer.  
↳shared-policer
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos egress rate-limit policer.  
↳stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification).

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos egress rate-limit policer.  
↳stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos egress rate-limit policer.  
↳stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification).

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos egress rate-limit policer.  
↳stats pass-excess-bytes
```

### drop-packets (state only)

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos egress rate-limit policer.  
↳stats drop-packets
```

### drop-bytes (state only)

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> qos egress rate-limit policer.  
↳stats drop-bytes
```

### counters (state only)

A collection of interface-related statistics objects.

### in-octets (state only)

The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> counters in-octets
```

### in-unicast-pkts (state only)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> counters in-unicast-pkts
```

### in-discards (state only)

The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> counters in-discards
```

### in-errors (state only)

For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> counters in-errors
```

### out-octets (state only)

The total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> counters out-octets
```

### out-unicast-pkts (state only)

The total number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> counters out-unicast-pkts
```

### out-discards (state only)

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> counters out-discards
```

### out-errors (state only)

For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface xvrf <xvrf> counters out-errors
```

## 3.2.23 qos

---

**Note:** requires a Turbo Router Network License.

---

QoS configuration.

```
vrouter running config# qos
```

### class-mask (config only)

Mask applied to marks.

```
vrouter running config# qos  
vrouter running qos# class-mask <0x0-0xffffffff>
```

### Default value

0xFFFFFFFF

**policer (config only)**

List of policer templates.

```
vrouter running config# qos policer <string>
```

|          |                        |
|----------|------------------------|
| <string> | Policer template name. |
|----------|------------------------|

**description (config only)**

A comment to describe the policer template.

```
vrouter running config# qos policer <string>
vrouter running policer <string># description <string>
```

**bandwidth (config only) (mandatory)**

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter running config# qos policer <string>
vrouter running policer <string># bandwidth BANDWIDTH
```

|                |  |
|----------------|--|
| BAND-<br>WIDTH | Rate in bits per second. K/M/G/T multipliers are supported. Example: 1G stands for 1000000000 bps. |
|----------------|--|

**burst (config only)**

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter running config# qos policer <string>
vrouter running policer <string># burst BURST
```

|       |  |
|-------|--|
| BURST | Burst size in bytes. K/M/G/T multipliers are supported. Example: 2K stands for 2000 bytes. |
|-------|--|

**excess-bandwidth (config only)**

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter running config# qos policer <string>
vrouter running policer <string># excess-bandwidth EXCESS-BANDWIDTH
```

|                      |  |
|----------------------|--|
| EXCESS-<br>BANDWIDTH | Rate in bits per second. K/M/G/T multipliers are supported. Example: 1G stands for 1000000000 bps. |
|----------------------|--|

**Default value**

0

**excess-burst (config only)**

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter running config# qos policer <string>
vrouter running policer <string># excess-burst EXCESS-BURST
```

|                  |  |
|------------------|--|
| EXCESS-<br>BURST | Burst size in bytes. K/M/G/T multipliers are supported. Example: 2K stands for 2000 bytes. |
|------------------|--|

**shared-policer**

List of shared policers.

```
vrouter running config# qos shared-policer <string>
```

|          |                      |
|----------|----------------------|
| <string> | Shared policer name. |
|----------|----------------------|

### description (config only)

A comment to describe the shared policer.

```
vrouter running config# qos shared-policer <string>  
vrouter running shared-policer <string># description <string>
```

### policer (config only)

Traffic policer template defined in the QoS context.

```
vrouter running config# qos shared-policer <string>  
vrouter running shared-policer <string># policer <leafref>
```

### bandwidth (state only)

Maximum bandwidth of regular traffic, a.k.a. CIR (Committed Information Rate), in bps. 0 allows no regular traffic.

```
vrouter> show state qos shared-policer <string> bandwidth
```

### burst (state only)

Maximum burst size of shaped traffic, a.k.a. CBS (Committed Burst Size), in bytes. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state qos shared-policer <string> burst
```

### excess-bandwidth (state only)

Maximum bandwidth of excess traffic, a.k.a. EIR (Excess Information Rate), in bps. 0 allows no excess traffic.

```
vrouter> show state qos shared-policer <string> excess-bandwidth
```

### excess-burst (state only)

Maximum burst size of excess traffic, a.k.a. EBS (Excess Burst Size), in bytes. The default value is set to excess-bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter> show state qos shared-policer <string> excess-burst
```

### stats (state only)

Traffic policer statistics.

### pass-packets (state only)

Number of packets passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouter> show state qos shared-policer <string> stats pass-packets
```

### pass-bytes (state only)

Number of bytes passed (regular traffic that conforms to (bandwidth, burst) specification.

```
vrouter> show state qos shared-policer <string> stats pass-bytes
```

### pass-excess-packets (state only)

Number of excess packets passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouter> show state qos shared-policer <string> stats pass-excess-packets
```

### pass-excess-bytes (state only)

Number of excess bytes passed (excess traffic that conforms to (excess-bandwidth, excess-burst) specification.

```
vrouter> show state qos shared-policer <string> stats pass-excess-bytes
```

**drop-packets (state only)**

Number of packets dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state qos shared-policer <string> stats drop-packets
```

**drop-bytes (state only)**

Number of bytes dropped (traffic that does not conform to bandwidth or excess-bandwidth).

```
vrouter> show state qos shared-policer <string> stats drop-bytes
```

**shaper (config only)**

List of shapers.

```
vrouter running config# qos shaper <string>
```

|          |              |
|----------|--------------|
| <string> | Shaper name. |
|----------|--------------|

**description (config only)**

A comment to describe the shaper.

```
vrouter running config# qos shaper <string>  
vrouter running shaper <string># description <string>
```

**bandwidth (config only) (mandatory)**

Maximum bandwidth of shaped traffic.

```
vrouter running config# qos shaper <string>  
vrouter running shaper <string># bandwidth BANDWIDTH
```

|                |  |
|----------------|--|
| BAND-<br>WIDTH | Rate in bits per second. K/M/G/T multipliers are supported. Example: 1G stands for 1000000000 bps. |
|----------------|--|

**burst (config only)**

Maximum burst size of shaped traffic. The default value is set to bandwidth / 80 to handle a burst of 100 ms at the targeted bandwidth. If not set or set to 0, the default value is applied.

```
vrouter running config# qos shaper <string>
vrouter running shaper <string># burst BURST
```

|       |  |
|-------|--|
| BURST | Burst size in bytes. K/M/G/T multipliers are supported. Example: 2K stands for 2000 bytes. |
|-------|--|

**layer1-overhead (config only)**

Number of bytes added by the underlying protocol on each packet.

```
vrouter running config# qos shaper <string>
vrouter running shaper <string># layer1-overhead <uint32>
```

**Default value**

0

**queue-size (config only)**

Number of packets that can be saved in the delay queue. If a scheduler is also configured on the interface, this value is not used, the queues of the scheduler are used as delay queues. The value is rounded up to the nearest power of 2.

```
vrouter running config# qos shaper <string>
vrouter running shaper <string># queue-size <uint32>
```

**Default value**

256

**scheduler (config only)**

List of schedulers.

```
vrouter running config# qos scheduler <string>
```

|          |                 |
|----------|-----------------|
| <string> | Scheduler name. |
|----------|-----------------|

### description (config only)

A comment to describe the scheduler.

```
vrouter running config# qos scheduler <string>  
vrouter running scheduler <string># description <string>
```

### core (config only)

Core assigned to manage the scheduler. If unset, cpu is automatically selected.

```
vrouter running config# qos scheduler <string>  
vrouter running scheduler <string># core <uint32>
```

### pq (config only)

Priority Queueing description.

```
vrouter running config# qos scheduler <string> pq
```

### nb-queue (config only) (mandatory)

Number of Priority Queueing queues available in the scheduler.

```
vrouter running config# qos scheduler <string> pq  
vrouter running pq# nb-queue <uint32>
```

### queue (config only)

List of Priority Queueing queues.

```
vrouter running config# qos scheduler <string> pq queue <uint32>
```

|          |                  |
|----------|------------------|
| <uint32> | Id of the queue. |
|----------|------------------|

### size (config only)

Size of the queue in packets.

```
vrouter running config# qos scheduler <string> pq queue <uint32>  
vrouter running queue <uint32># size <uint32>
```

#### Default value

256

### policer (config only)

Traffic policer defined in the QoS context applied to incoming traffic.

```
vrouter running config# qos scheduler <string> pq queue <uint32>  
vrouter running queue <uint32># policer <leafref>
```

### shaper (config only)

Traffic shaper defined in the QoS context applied to outgoing traffic.

```
vrouter running config# qos scheduler <string> pq queue <uint32>  
vrouter running queue <uint32># shaper <leafref>
```

### class (config only)

List of traffic classes bound to this queue.

```
vrouter running config# qos scheduler <string> pq queue <uint32>  
vrouter running queue <uint32># class <leafref>
```

|           |             |
|-----------|-------------|
| <leafref> | Class name. |
|-----------|-------------|

**pb-dwrr (config only)**

Priority-Based Deficit Weighted Round Robin description.

```
vrouter running config# qos scheduler <string> pb-dwrr
```

**nb-queue (config only) (mandatory)**

Number of PB-DWRR queues available in the scheduler.

```
vrouter running config# qos scheduler <string> pb-dwrr
vrouter running pb-dwrr# nb-queue <uint32>
```

**queue (config only)**

List of PB-DWRR queues.

```
vrouter running config# qos scheduler <string> pb-dwrr queue <uint32>
```

|          |                  |
|----------|------------------|
| <uint32> | Id of the queue. |
|----------|------------------|

**size (config only)**

Size of the queue in packets.

```
vrouter running config# qos scheduler <string> pb-dwrr queue <uint32>
vrouter running queue <uint32># size <uint32>
```

**Default value**

256

**policer (config only)**

Traffic policer defined in the QoS context applied to incoming traffic.

```
vrouter running config# qos scheduler <string> pb-dwrr queue <uint32>
vrouter running queue <uint32># policer <leafref>
```

### shaper (config only)

Traffic shaper defined in the QoS context applied to outgoing traffic.

```
vrouter running config# qos scheduler <string> pb-dwrr queue <uint32>
vrouter running queue <uint32># shaper <leafref>
```

### quantum (config only)

Quantum of the queue. Relevant only if priority is low.

```
vrouter running config# qos scheduler <string> pb-dwrr queue <uint32>
vrouter running queue <uint32># quantum <uint32>
```

#### Default value

1500

### priority (config only)

Priority of the queue (low or high).

```
vrouter running config# qos scheduler <string> pb-dwrr queue <uint32>
vrouter running queue <uint32># priority PRIORITY
```

| PRIORITY values | Description    |
|-----------------|----------------|
| low             | Low priority.  |
| high            | High priority. |

#### Default value

low

### class (config only)

List of traffic classes bound to this queue.

```
vrouter running config# qos scheduler <string> pb-dwrr queue <uint32>
vrouter running queue <uint32># class <leafref>
```

|           |             |
|-----------|-------------|
| <leafref> | Class name. |
|-----------|-------------|

**class (config only)**

List of supported classes.

```
vrouter running config# qos class <string>
```

|          |             |
|----------|-------------|
| <string> | Class name. |
|----------|-------------|

**description (config only)**

A comment to describe the class.

```
vrouter running config# qos class <string>
vrouter running class <string># description <string>
```

**mark (config only)**

Class mark. Optional if cp is true.

```
vrouter running config# qos class <string>
vrouter running class <string># mark <0x0-0xffffffff>
```

**cp (config only)**

Whether this class relates to critical control plane traffic. If unset, match any traffic. If true, only match critical control plane traffic. If false, do not match critical control plane traffic.

```
vrouter running config# qos class <string>
vrouter running class <string># cp true|false
```

**3.2.24 vrrp****global**


---

**Note:** requires a Turbo Router Network License.

---

Virtual Router Redundancy Protocol service.

```
vrouter running config# vrf <vrf> vrrp
```

### enabled

Enable or disable the VRRP service.

```
vrouter running config# vrf <vrf> vrrp
vrouter running vrrp# enabled true|false
```

### Default value

true

### router-id

String identifying the machine.

```
vrouter running config# vrf <vrf> vrrp
vrouter running vrrp# router-id <string>
```

### Default value

router

### traps-enabled

Enable or disable SNMP traps.

```
vrouter running config# vrf <vrf> vrrp
vrouter running vrrp# traps-enabled true|false
```

### Default value

false

### vrrp-startup-delay

Delay in seconds before vrrp instances start up after keepalived starts. Recommended value is 30 when at least one of the vrrp instance runs on top of lag interfaces.

```
vrouter running config# vrf <vrf> vrrp
vrouter running vrrp# vrrp-startup-delay <uint16>
```

**Default value**

0

**group**

Group of VRRP instances that change state together.

```
vrouter running config# vrf <vrf> vrrp group <string>
```

|          |                  |
|----------|------------------|
| <string> | VRRP group name. |
|----------|------------------|

**instance**

List of VRRP instances in this group. All instances of a same group share their state.

```
vrouter running config# vrf <vrf> vrrp group <string>  
vrouter running group <string># instance <leafref>
```

**notify-ha-group**

Associate the VRRP group to a high-availability group to notify VRRP state.

```
vrouter running config# vrf <vrf> vrrp group <string>  
vrouter running group <string># notify-ha-group <leafref>
```

**state (state only)**

VRRP group state.

```
vrouter> show state vrf <vrf> vrrp group <string> state
```

**interface**

---

**Note:** requires a Turbo Router Network License.

---

The list of VRRP interfaces on the device.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
```

|        |                    |
|--------|--------------------|
| <vrrp> | An interface name. |
|--------|--------------------|

### mtu

Set the max transmission unit size in octets.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>  
vrouter running vrrp <vrrp># mtu <uint32>
```

### promiscuous

Set promiscuous mode.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>  
vrouter running vrrp <vrrp># promiscuous true|false
```

### description

A textual description of the interface.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>  
vrouter running vrrp <vrrp># description <string>
```

### enabled

The desired (administrative) state of the interface.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>  
vrouter running vrrp <vrrp># enabled true|false
```

### Default value

true

## version

VRRP version 2 for IPv4, 3 for IPv4 or IPv6.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># version <uint8>
```

### Default value

2

## link-interface (mandatory)

The interface bound by VRRP.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># link-interface LINK-INTERFACE
```

|                |                    |
|----------------|--------------------|
| LINK-INTERFACE | An interface name. |
|----------------|--------------------|

## garp-delay

Delay for the second set of gratuitous ARP after transition to master state.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># garp-delay <uint16>
```

### Default value

5

## use-vmac

If true, create and associate the virtual address to a vmac interface for this VRRP instance with a VRRP standard MAC address.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># use-vmac true|false
```

### Default value

true

### vmac-xmit-base

If true, send and receive VRRP messages from bound interface instead of VMAC interface. It requires use-vmac to be set to true.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># vmac-xmit-base true|false
```

#### Default value

false

### vrid (mandatory)

Virtual router identifier, used to differentiate multiple VRRP instances bound to the same interface.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># vrid <uint8>
```

### priority

Specifies the sending VRRP interface's priority for the virtual router. The higher value among interfaces with the same router id will be elected as master.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># priority <uint8>
```

#### Default value

100

### init-state

Initial VRRP state.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># init-state INIT-STATE
```

| INIT-STATE values | Description  |
|-------------------|--|
| master            | Master state: the router functions as the forwarding router (rfc5798#6.4.3).           |
| backup            | Backup state: monitor the availability and state of the Master Router (rfc5798#6.4.2). |

#### Default value

backup

### preempt

If true, the VRRP instance becomes master when lower priority advertisements are received from the other router. For this to work, the initial state of this entry must be backup.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>  
vrouter running vrrp <vrrp># preempt true|false
```

#### Default value

true

### preempt-delay

Additional delay the router waits before preempting the master state after receiving a lower priority advertisements from another node. A value of 0 does not mean immediate switchover, as it is still delayed by Master\_Down\_Interval.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>  
vrouter running vrrp <vrrp># preempt-delay <uint16>
```

#### Default value

0

### advertisement-interval

Interval between successive VRRP advertisements in milliseconds.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>  
vrouter running vrrp <vrrp># advertisement-interval <uint16>
```

#### Default value

1000

### track-link-interface

If false, the VRRP instance (and its group if any) does not go to fault state if the link-interface state goes down. Set to false to prevent a broken ha link from causing a fault state on both nodes.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># track-link-interface true|false
```

#### Default value

true

### track-interface

List of tracked interfaces. The VRRP instance (and its group if any) goes to fault state if one of the tracked interfaces goes down.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># track-interface TRACK-INTERFACE
```

|                 |                    |
|-----------------|--------------------|
| TRACK-INTERFACE | An interface name. |
|-----------------|--------------------|

### track

A tracker name. The VRRP instance (and its group if any) goes to fault state if the tracker is down.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># track TRACK
```

|       |                  |
|-------|------------------|
| TRACK | An tracker name. |
|-------|------------------|

### track-fast-path

If true, the VRRP instance (and its group if any) goes to fault state if fast path state does not match the configuration.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># track-fast-path true|false
```

#### Default value

false

## notify-ha-group

Associate the VRRP instance to a high-availability group to notify VRRP state.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>  
vrouter running vrrp <vrrp># notify-ha-group <leafref>
```

## ifindex (state only)

System assigned number for each interface. Corresponds to ifIndex object in SNMP Interface MIB.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ifindex
```

## admin-status (state only)

The desired state of the interface. In RFC 7223 this leaf has the same read semantics as ifAdminStatus. Here, it reflects the administrative state as set by enabling or disabling the interface.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> admin-status
```

## oper-status (state only)

The current operational state of the interface. This leaf has the same semantics as ifOperStatus.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> oper-status
```

## last-change (state only)

This timestamp indicates the time of the last state change of the interface (e.g., up-to-down transition). This corresponds to the ifLastChange object in the standard interface MIB. The value is the timestamp in nanoseconds relative to the Unix Epoch (Jan 1, 1970 00:00:00 UTC).

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> last-change
```

### state (state only)

Current VRRP state.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> state
```

### network-stack

Network stack parameters for this interface.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack
```

### ipv4

IPv4 parameters.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv4
```

### send-redirects

Send ICMP redirect if host is on the same network than gateway.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv4  
vrouter running ipv4# send-redirects true|false
```

### accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv4  
vrouter running ipv4# accept-redirects true|false
```

### accept-source-route

Accept packets with source route option. Must be activated at vrf or system level too to be activated.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv4
vrouter running ipv4# accept-source-route true|false
```

### arp-announce

Define different restriction levels for announcing the local source IP address from IP packets in ARP requests sent on interface. Increasing the restriction level gives more chance for receiving answer from the resolved target while decreasing the level announces more valid sender’s information.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv4
vrouter running ipv4# arp-announce ARP-ANNOUNCE
```

| ARP-ANNOUNCE values | Description  |
|---------------------|--|
| any                 | Use any local address, configured on any interface.  |
| avoid-not-in-subnet | Try to avoid local addresses that are not in the target’s subnet for this interface. This mode is useful when target hosts reachable via this interface require the source IP address in ARP requests to be part of their logical network configured on the receiving interface. When we generate the request we will check all our subnets that include the target IP and will preserve the source address if it is from such subnet. If there is no such subnet we select source address according to the rules for level 2, ‘best-local’.   |
| best-local          | Always use the best local address for this target. In this mode we ignore the source address in the IP packet and try to select local address that we prefer for talks with the target host. Such local address is selected by looking for primary IP addresses on all our subnets on the outgoing interface that include the target IP address. If no suitable local address is found we select the first local address we have on the outgoing interface or on all other interfaces, with the hope we will receive reply for our request and even sometimes no matter the source IP address we announce. |

### arp-filter

Allows to have multiple network interfaces on the same subnet, and have the ARPs for each interface be answered based on whether or not the kernel would route a packet from the ARP’d IP out that interface (therefore you must use source based routing for this to work). In other words it allows control of which cards (usually 1) will respond to an arp request.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv4
vrouter running ipv4# arp-filter true|false
```

## arp-ignore

Define different modes for sending replies in response to received ARP requests that resolve local target IP addresses.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv4
vrouter running ipv4# arp-ignore ARP-IGNORE
```

| ARP-IGNORE values          | Description  |
|----------------------------|--|
| any                        | Reply for any local target IP address, configured on any interface.  |
| check-interface            | Reply only if the target IP address is local address configured on the incoming interface.   |
| check-interface-and-subnet | Reply only if the target IP address is local address configured on the incoming interface and both with the sender's IP address are part from same subnet on this interface. |
| ignore-scope               | Do not reply for local addresses configured with scope host, only resolutions for global and link addresses are replied.   |
| ignore-all                 | Do not reply for all local addresses.  |

## log-invalid-addresses

Log packets with impossible addresses.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv4
vrouter running ipv4# log-invalid-addresses true|false
```

## ipv6

IPv6 parameters.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv6
```

## autoconfiguration

Autoconfigure addresses using Prefix Information in Router Advertisements.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv6
vrouter running ipv6# autoconfiguration true|false
```

## accept-router-advert

Accept Router Advertisements.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv6
vrouter running ipv6# accept-router-advert ACCEPT-ROUTER-ADVERT
```

| ACCEPT-ROUTER-ADVERT values | Description   |
|-----------------------------|---|
| never                       | Do not accept Router Advertisements.                        |
| norouter-mode               | Accept Router Advertisements if forwarding is disabled.     |
| always                      | Accept Router Advertisements even if forwarding is enabled. |

## accept-redirects

Accept redirect when acting as a host. It is always disabled when acting as a router.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv6
vrouter running ipv6# accept-redirects true|false
```

## accept-source-route

Accept packets with source route option.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv6
vrouter running ipv6# accept-source-route true|false
```

## router-solicitations

Number of Router Solicitations to send until assuming no routers are present.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv6
vrouter running ipv6# router-solicitations <int16>
```

## use-temporary-addresses

Preference for Privacy Extensions (RFC4941). Not applied to point-to-point and loopback devices (always 0).

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> network-stack ipv6
vrouter running ipv6# use-temporary-addresses USE-TEMPORARY-ADDRESSES
```

| USE-TEMPORARY-ADDRESSES values | Description   |
|--------------------------------|---|
| never                          | Disable Privacy Extensions, i.e. use the public address, subnet prefix/interface id, where interface id is always the same. |
| prefer-public-addresses        | Enable Privacy Extensions, but prefer public addresses over temporary addresses.  |
| always                         | Enable Privacy Extensions and prefer temporary addresses over public addresses.   |

## authentication

Authentication parameters.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> authentication
```

## auth-type

Authentication type: password or IPsec. Authentication is disabled if unset.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> authentication
vrouter running authentication# auth-type AUTH-TYPE
```

| AUTH-TYPE values | Description |
|------------------|-------------|
| pass             | Password.   |
| ah               | AH.         |

## auth-pass

VRRP password. It should be the same on all VRRP instances.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp> authentication
vrouter running authentication# auth-pass <string>
```

## unicast-peer

IP addresses of unicast peers. If the list is not empty, do not send VRRP advertisements over a VRRP multicast group but to this list of peers.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># unicast-peer <unicast-peer>
```

| <unicast-peer> values | Description      |
|-----------------------|------------------|
| <A.B.C.D>             | An IPv4 address. |
| <X:X::X:X>            | An IPv6 address. |

## virtual-address

IP addresses added on master switch and deleted on backup switch.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># virtual-address <virtual-address>
```

| <virtual-address> values | Description   |
|--------------------------|---|
| <A.B.C.D/M>              | A masked IPv4 address: address and prefix of that subnet. |
| <X:X::X:X/M>             | A masked IPv6 address: address and prefix of that subnet. |

## virtual-route

Routes added on master switch and deleted on backup switch.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># virtual-route <virtual-route> interface <string> \
... gw GW
```

| <virtual-route> values | Description                            |
|------------------------|--|
| <A.B.C.D/M>            | An IPv4 prefix: address and CIDR mask. |
| <X:X::X:X/M>           | An IPv6 prefix: address and CIDR mask. |

## interface

Out device.

```
interface <string>
```

## gw

Gateway IP.

```
gw GW
```

| GW values  | Description      |
|------------|------------------|
| <A.B.C.D>  | An IPv4 address. |
| <X:X::X:X> | An IPv6 address. |

## counters (state only)

A collection of interface-related statistics objects.

## in-octets (state only)

The total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> counters in-octets
```

## in-unicast-pkts (state only)

The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were not addressed to a multicast or broadcast address at this sub-layer. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> counters in-unicast-pkts
```

### in-discards (state only)

The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> counters in-discards
```

### in-errors (state only)

For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> counters in-errors
```

### out-octets (state only)

The total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> counters out-octets
```

### out-unicast-pkts (state only)

The total number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> counters out-unicast-pkts
```

### out-discards (state only)

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> counters out-discards
```

### out-errors (state only)

For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'last-clear'.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> counters out-errors
```

### ipv4 (state only)

Parameters for the IPv4 address family.

### enabled (state only)

Controls whether IPv4 is enabled or disabled on this interface. When IPv4 is enabled, this interface is connected to an IPv4 stack, and the interface can send and receive IPv4 packets.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 enabled
```

### address (state only)

The list of configured IPv4 addresses on the interface.

**peer (state only)**

The IPv4 address of the remote endpoint for point to point interfaces.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 address <address> peer
```

**origin (state only)**

The origin of this address, e.g., statically configured, assigned by DHCP, etc..

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 address <address> origin
```

**neighbor (state only)**

A list of mappings from IPv4 addresses to link-layer addresses. Entries in this list are used as static entries in the ARP Cache.

**link-layer-address (state only)**

The link-layer address of the neighbor node.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 neighbor <neighbor> link-  
↪layer-address
```

**state (state only)**

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 neighbor <neighbor> state
```

**dhcp (state only)**

DHCP client configuration.

**enabled (state only)**

Enable or disable DHCP.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp enabled
```

**timeout (state only)**

Time before deciding that it's not going to be able to contact a server.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp timeout
```

**retry (state only)**

Time before trying again to contact a DHCP server.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp retry
```

**select-timeout (state only)**

Time at which the client stops waiting for other offers from servers.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp select-timeout
```

**reboot (state only)**

Time after trying to reacquire its old address before trying to discover a new address.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp reboot
```

**initial-interval (state only)**

Time between the first attempt to reach a server and the second attempt to reach a server.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp initial-interval
```

### dhcp-lease-time (state only)

Requested lease time.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp dhcp-lease-time
```

### dhcp-client-identifier-ascii (state only)

DHCP client identifier (ASCII).

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp dhcp-client-identifier-  
↪ascii
```

### dhcp-client-identifier-hexa (state only)

DHCP client identifier (hexadecimal).

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp dhcp-client-identifier-  
↪hexa
```

### host-name (state only)

DHCP client name.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp host-name
```

### request (state only)

DHCP requests.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp request
```

### current-lease (state only)

Current lease.

**fixed-address (state only)**

The IPv4 address on the interface.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp current-lease fixed-  
↪address
```

**renew (state only)**

Time at which the client should begin trying to contact its server to renew its lease.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp current-lease renew
```

**rebind (state only)**

Time at which the client should begin to try to contact any dhcp server to renew its lease.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp current-lease rebind
```

**expire (state only)**

Time at which the client must stop using a lease if it has not been able to renew it.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv4 dhcp current-lease expire
```

**ipv6 (state only)**

Parameters for the IPv6 address family.

**enabled (state only)**

Controls whether IPv6 is enabled or disabled on this interface. When IPv6 is enabled, this interface is connected to an IPv6 stack, and the interface can send and receive IPv6 packets.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv6 enabled
```

**address (state only)**

The list of configured IPv6 addresses on the interface.

**peer (state only)**

The IPv6 address of the remote endpoint for point to point interfaces.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv6 address <address> peer
```

**origin (state only)**

The origin of this address, e.g., static, dhcp, etc.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv6 address <address> origin
```

**status (state only)**

The status of an address. Most of the states correspond to states from the IPv6 Stateless Address Autoconfiguration protocol.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv6 address <address> status
```

**neighbor (state only)**

List of IPv6 neighbors.

**link-layer-address (state only)**

The link-layer address of the neighbor node.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv6 neighbor <neighbor> link-  
↪layer-address
```

### router (state only)

Indicates that the neighbor node acts as a router.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv6 neighbor <neighbor> router
```

### state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ipv6 neighbor <neighbor> state
```

### ethernet (state only)

Top-level container for Ethernet state.

### mac-address (state only)

MAC address assigned to the Ethernet interface.

```
vrouter> show state vrf <vrf> interface vrrp <vrrp> ethernet mac-address
```

## 3.2.25 ike

---

**Note:** requires a Turbo IPsec Application License.

---

IKE configuration.

```
vrouter running config# vrf <vrf> ike
```

### enabled

Enable or disable the IKE protocol and indicate whether the system should negotiate Security Associations for the IPsec protocol.

```
vrouter running config# vrf <vrf> ike  
vrouter running ike# enabled true|false
```

### Default value

true

### pool

List of virtual address pools.

```
vrouter running config# vrf <vrf> ike pool <pool>
```

|        |                       |
|--------|-----------------------|
| <pool> | IKE object name type. |
|--------|-----------------------|

### address (mandatory)

Virtual addresses in the pool.

```
vrouter running config# vrf <vrf> ike pool <pool>
vrouter running pool <pool># address ADDRESS
```

| ADDRESS values | Description                                     |
|----------------|---|
| <ipv4-address> | An IPv4 address.                                |
| <ipv6-address> | An IPv6 address.                                |
| <ipv4-prefix>  | An IPv4 prefix: address and CIDR mask.          |
| <ipv6-prefix>  | An IPv6 prefix: address and CIDR mask.          |
| <ipv4-range>   | An IPv4 address range, in the form addr4-addr4. |
| <ipv6-range>   | An IPv6 address range, in the form addr6-addr6. |

### dns

List of DNS (Domain Name Service) servers IP addresses.

```
vrouter running config# vrf <vrf> ike pool <pool>
vrouter running pool <pool># dns DNS
```

| DNS values     | Description      |
|----------------|------------------|
| <ipv4-address> | An IPv4 address. |
| <ipv6-address> | An IPv6 address. |

## nbns

List of NBNS (NetBIOS Name Service) servers IP addresses.

```
vrouter running config# vrf <vrf> ike pool <pool>
vrouter running pool <pool># nbns NBNS
```

| NBNS values    | Description      |
|----------------|------------------|
| <ipv4-address> | An IPv4 address. |
| <ipv6-address> | An IPv6 address. |

## dhcp

List of DHCP servers IP addresses.

```
vrouter running config# vrf <vrf> ike pool <pool>
vrouter running pool <pool># dhcp DHCP
```

| DHCP values    | Description      |
|----------------|------------------|
| <ipv4-address> | An IPv4 address. |
| <ipv6-address> | An IPv6 address. |

## subnet

List of sub-networks that this device protects (attributes INTERNAL\_IP4\_SUBNET/INTERNAL\_IP6\_SUBNET).

```
vrouter running config# vrf <vrf> ike pool <pool>
vrouter running pool <pool># subnet SUBNET
```

| SUBNET<br>val-<br>ues   | Description  |
|-------------------------|--|
| <subnet ip-<br>address> | The ipv4-prefix type represents an IPv4 address prefix. The prefix length is given by the number following the slash character and must be less than or equal to 32. A prefix length value of n corresponds to an IP address mask that has n contiguous 1-bits from the most significant bit (MSB) and all other bits set to 0. The canonical format of an IPv4 prefix has all bits of the IPv4 address set to zero that are not part of the IPv4 prefix.  |
| <subnet ip-<br>address> | The ipv6-prefix type represents an IPv6 address prefix. The prefix length is given by the number following the slash character and must be less than or equal to 128. A prefix length value of n corresponds to an IP address mask that has n contiguous 1-bits from the most significant bit (MSB) and all other bits set to 0. The IPv6 address should have all bits that do not belong to the prefix set to zero. The canonical format of an IPv6 prefix has all bits of the IPv6 address set to zero that are not part of the IPv6 prefix. Furthermore, the IPv6 address is represented as defined in Section 4 of RFC 5952. |

### certificate

List of X509 certificates.

```
vrouter running config# vrf <vrf> ike certificate <certificate>
```

|               |                       |
|---------------|-----------------------|
| <certificate> | IKE object name type. |
|---------------|-----------------------|

### certificate (mandatory)

PEM-encoded X509 certificate.

```
vrouter running config# vrf <vrf> ike certificate <certificate>
vrouter running certificate <certificate># certificate <string>
```

### private-key (mandatory)

PEM-encoded X509 private key.

```
vrouter running config# vrf <vrf> ike certificate <certificate>
vrouter running certificate <certificate># private-key <string>
```

### certificate-authority

List of X509 CA certificates.

```
vrouter running config# vrf <vrf> ike certificate-authority <certificate-authority>
```

|                         |                       |
|-------------------------|-----------------------|
| <certificate-authority> | IKE object name type. |
|-------------------------|-----------------------|

### certificate (mandatory)

PEM-encoded X509 certificate.

```
vrouter running config# vrf <vrf> ike certificate-authority <certificate-authority>
vrouter running certificate-authority <certificate-authority># certificate <string>
```

### crl

PEM-encoded X509 certificate revocation list.

```
vrouter running config# vrf <vrf> ike certificate-authority <certificate-authority>
vrouter running certificate-authority <certificate-authority># crl <string>
```

### crl-uri

List of CRL distribution points (ldap or http URIs).

```
vrouter running config# vrf <vrf> ike certificate-authority <certificate-authority>
vrouter running certificate-authority <certificate-authority># crl-uri CRL-URI
```

|         |  |
|---------|--|
| CRL-URI | An ASCII-encoded Uniform Resource Identifier (URI) as defined in RFC 3986. |
|---------|--|

### pre-shared-key

List of pre-shared keys.

```
vrouter running config# vrf <vrf> ike pre-shared-key <pre-shared-key>
```

|                  |                       |
|------------------|-----------------------|
| <pre-shared-key> | IKE object name type. |
|------------------|-----------------------|

## id

List of IKE identities the IKE pre-shared secret belongs to.

```
vrouter running config# vrf <vrf> ike pre-shared-key <pre-shared-key>
vrouter running pre-shared-key <pre-shared-key># id ID
```

| ID values | Description  |
|-----------|--|
| <ike-id>  | An IPv4 address.   |
| <ike-id>  | An IPv6 address.   |
| <ike-id>  | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <ike-id>  | IKE ID (IP address, fqdn, e-mail address or distinguished name).   |
| <ike-id>  | IKE ID (IP address, fqdn, e-mail address or distinguished name).   |

## secret (mandatory)

Value of the IKE pre-shared secret.

```
vrouter running config# vrf <vrf> ike pre-shared-key <pre-shared-key>
vrouter running pre-shared-key <pre-shared-key># secret SECRET
```

| SECRET values      | Description            |
|--------------------|------------------------|
| <0x-hex-string>    | Pre-shared key secret. |
| <0s-base64-string> | Pre-shared key secret. |
| <ascii-string>     | Pre-shared key secret. |

## eap-key

List of EAP keys.

```
vrouter running config# vrf <vrf> ike eap-key <eap-key>
```

|           |                       |
|-----------|-----------------------|
| <eap-key> | IKE object name type. |
|-----------|-----------------------|

## id

List of EAP identities the EAP secret belongs to.

```
vrouter running config# vrf <vrf> ike eap-key <eap-key>
vrouter running eap-key <eap-key># id ID
```

|    |         |
|----|---------|
| ID | EAP ID. |
|----|---------|

## secret (mandatory)

Value of the EAP secret.

```
vrouter running config# vrf <vrf> ike eap-key <eap-key>
vrouter running eap-key <eap-key># secret SECRET
```

| SECRET values      | Description            |
|--------------------|------------------------|
| <0x-hex-string>    | Pre-shared key secret. |
| <0s-base64-string> | Pre-shared key secret. |
| <ascii-string>     | Pre-shared key secret. |

## eap-radius

EAP RADIUS parameters.

```
vrouter running config# vrf <vrf> ike eap-radius
```

### nas-identifier

Network Access Server identifier.

```
vrouter running config# vrf <vrf> ike eap-radius  
vrouter running eap-radius# nas-identifier <string>
```

#### Default value

6WINDvRouter

### auth-port

RADIUS server port number for EAP authentication.

```
vrouter running config# vrf <vrf> ike eap-radius  
vrouter running eap-radius# auth-port <uint16>
```

#### Default value

1812

### sockets

Maximum simultaneous authentication sessions with the RADIUS server.

```
vrouter running config# vrf <vrf> ike eap-radius  
vrouter running eap-radius# sockets <uint32>
```

#### Default value

1

### retransmit-tries

Number of times to retransmit a packet before giving up.

```
vrouter running config# vrf <vrf> ike eap-radius  
vrouter running eap-radius# retransmit-tries <0..100>
```

#### Default value

4

## retransmit-timeout

Timeout in seconds before sending first retransmit.

```
vrouter running config# vrf <vrf> ike eap-radius
vrouter running eap-radius# retransmit-timeout <0.000 .. 60.000>
```

### Default value

2.0

## retransmit-base

Base to use for calculating retransmit exponential back off.

```
vrouter running config# vrf <vrf> ike eap-radius
vrouter running eap-radius# retransmit-base <0.000 .. 10.000>
```

### Default value

1.4

## server

List of RADIUS servers for EAP.

```
vrouter running config# vrf <vrf> ike eap-radius server <server>
```

|          |                       |
|----------|-----------------------|
| <server> | IKE object name type. |
|----------|-----------------------|

## address (mandatory)

RADIUS server IP address.

```
vrouter running config# vrf <vrf> ike eap-radius server <server>
vrouter running server <server># address ADDRESS
```

| ADDRESS values | Description      |
|----------------|------------------|
| <ipv4-address> | An IPv4 address. |
| <ipv6-address> | An IPv6 address. |

**secret (mandatory)**

Secret shared with the RADIUS server.

```
vrouter running config# vrf <vrf> ike eap-radius server <server>
vrouter running server <server># secret SECRET
```

| SECRET values      | Description            |
|--------------------|------------------------|
| <0x-hex-string>    | Pre-shared key secret. |
| <0s-base64-string> | Pre-shared key secret. |
| <ascii-string>     | Pre-shared key secret. |

**nas-identifier**

Network Access Server identifier.

```
vrouter running config# vrf <vrf> ike eap-radius server <server>
vrouter running server <server># nas-identifier <string>
```

**auth-port**

RADIUS server port number for EAP authentication.

```
vrouter running config# vrf <vrf> ike eap-radius server <server>
vrouter running server <server># auth-port <uint16>
```

**sockets**

Maximum simultaneous authentication sessions with the RADIUS server.

```
vrouter running config# vrf <vrf> ike eap-radius server <server>
vrouter running server <server># sockets <uint32>
```

### retransmit-tries

Number of times to retransmit a packet before giving up.

```
vrouter running config# vrf <vrf> ike eap-radius server <server>  
vrouter running server <server># retransmit-tries <0..100>
```

### retransmit-timeout

Timeout in seconds before sending first retransmit.

```
vrouter running config# vrf <vrf> ike eap-radius server <server>  
vrouter running server <server># retransmit-timeout <0.000 .. 60.000>
```

### retransmit-base

Base to use for calculating retransmit exponential back off.

```
vrouter running config# vrf <vrf> ike eap-radius server <server>  
vrouter running server <server># retransmit-base <0.000 .. 10.000>
```

### logging

Logs configuration.

```
vrouter running config# vrf <vrf> ike logging
```

### daemon

Max level of messages logged in the system daemons facility.

```
vrouter running config# vrf <vrf> ike logging daemon
```

## default

Default max log level.

```
vrouter running config# vrf <vrf> ike logging daemon
vrouter running daemon# default DEFAULT
```

| DEFAULT values | Description   |
|----------------|---|
| disable        | No log.   |
| 0              | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1              | Generic control flow with errors, a good default to see whats going on. |
| 2              | More detailed debugging control flow.                                   |
| 3              | Including RAW data dumps in hex.  |
| 4              | Also include sensitive material in dumps, e.g. keys.                    |

### Default value

0

## asn1

Low-level encoding/decoding (ASN.1, X.509 etc.).

```
vrouter running config# vrf <vrf> ike logging daemon
vrouter running daemon# asn1 ASN1
```

| ASN1 values | Description   |
|-------------|---|
| disable     | No log.   |
| 0           | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1           | Generic control flow with errors, a good default to see whats going on. |
| 2           | More detailed debugging control flow.                                   |
| 3           | Including RAW data dumps in hex.  |
| 4           | Also include sensitive material in dumps, e.g. keys.                    |

## config

Configuration management and plugins.

```
vrouter running config# vrf <vrf> ike logging daemon
vrouter running daemon# config CONFIG
```

| CONFIG values | Description   |
|---------------|---|
| disable       | No log.   |
| 0             | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1             | Generic control flow with errors, a good default to see whats going on. |
| 2             | More detailed debugging control flow.                                   |
| 3             | Including RAW data dumps in hex.  |
| 4             | Also include sensitive material in dumps, e.g. keys.                    |

## child

CHILD\_SA/IPsec SA processing.

```
vrouter running config# vrf <vrf> ike logging daemon
vrouter running daemon# child CHILD
```

| CHILD values | Description   |
|--------------|---|
| disable      | No log.   |
| 0            | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1            | Generic control flow with errors, a good default to see whats going on. |
| 2            | More detailed debugging control flow.                                   |
| 3            | Including RAW data dumps in hex.  |
| 4            | Also include sensitive material in dumps, e.g. keys.                    |

## daemon

Main daemon setup/cleanup/signal handling.

```
vrouter running config# vrf <vrf> ike logging daemon
vrouter running daemon# daemon DAEMON
```

| DAEMON values | Description   |
|---------------|---|
| disable       | No log.   |
| 0             | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1             | Generic control flow with errors, a good default to see whats going on. |
| 2             | More detailed debugging control flow.                                   |
| 3             | Including RAW data dumps in hex.  |
| 4             | Also include sensitive material in dumps, e.g. keys.                    |

## encoding

Packet encoding/decoding encryption/decryption operations.

```
vrouter running config# vrf <vrf> ike logging daemon
vrouter running daemon# encoding ENCODING
```

| ENCODING values | Description   |
|-----------------|---|
| disable         | No log.   |
| 0               | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1               | Generic control flow with errors, a good default to see whats going on. |
| 2               | More detailed debugging control flow.                                   |
| 3               | Including RAW data dumps in hex.  |
| 4               | Also include sensitive material in dumps, e.g. keys.                    |

## ipsec

Libipsec library messages.

```
vrouter running config# vrf <vrf> ike logging daemon
vrouter running daemon# ipsec IPSEC
```

| IPSEC values | Description   |
|--------------|---|
| disable      | No log.   |
| 0            | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1            | Generic control flow with errors, a good default to see whats going on. |
| 2            | More detailed debugging control flow.                                   |
| 3            | Including RAW data dumps in hex.  |
| 4            | Also include sensitive material in dumps, e.g. keys.                    |

## ike

IKE\_SA/ISAKMP SA processing.

```
vrouter running config# vrf <vrf> ike logging daemon
vrouter running daemon# ike IKE
```

| IKE values | Description   |
|------------|---|
| disable    | No log.   |
| 0          | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1          | Generic control flow with errors, a good default to see whats going on. |
| 2          | More detailed debugging control flow.                                   |
| 3          | Including RAW data dumps in hex.  |
| 4          | Also include sensitive material in dumps, e.g. keys.                    |

## job

Jobs queuing/processing and thread pool management.

```
vrouter running config# vrf <vrf> ike logging daemon
vrouter running daemon# job JOB
```

| JOB values | Description   |
|------------|---|
| disable    | No log.   |
| 0          | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1          | Generic control flow with errors, a good default to see whats going on. |
| 2          | More detailed debugging control flow.                                   |
| 3          | Including RAW data dumps in hex.  |
| 4          | Also include sensitive material in dumps, e.g. keys.                    |

## kernel

IPsec/Networking kernel interface.

```
vrouter running config# vrf <vrf> ike logging daemon
vrouter running daemon# kernel KERNEL
```

| KERNEL values | Description   |
|---------------|---|
| disable       | No log.   |
| 0             | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1             | Generic control flow with errors, a good default to see whats going on. |
| 2             | More detailed debugging control flow.                                   |
| 3             | Including RAW data dumps in hex.  |
| 4             | Also include sensitive material in dumps, e.g. keys.                    |

## library

Libstrongwan library messages.

```
vrouter running config# vrf <vrf> ike logging daemon
vrouter running daemon# library LIBRARY
```

| LIBRARY values | Description   |
|----------------|---|
| disable        | No log.   |
| 0              | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1              | Generic control flow with errors, a good default to see whats going on. |
| 2              | More detailed debugging control flow.                                   |
| 3              | Including RAW data dumps in hex.  |
| 4              | Also include sensitive material in dumps, e.g. keys.                    |

## manager

IKE\_SA manager, handling synchronization for IKE\_SA access.

```
vrouter running config# vrf <vrf> ike logging daemon
vrouter running daemon# manager MANAGER
```

| MANAGER values | Description   |
|----------------|---|
| disable        | No log.   |
| 0              | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1              | Generic control flow with errors, a good default to see whats going on. |
| 2              | More detailed debugging control flow.                                   |
| 3              | Including RAW data dumps in hex.  |
| 4              | Also include sensitive material in dumps, e.g. keys.                    |

## network

IKE network communication.

```
vrouter running config# vrf <vrf> ike logging daemon
vrouter running daemon# network NETWORK
```

| NETWORK values | Description   |
|----------------|---|
| disable        | No log.   |
| 0              | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1              | Generic control flow with errors, a good default to see whats going on. |
| 2              | More detailed debugging control flow.                                   |
| 3              | Including RAW data dumps in hex.  |
| 4              | Also include sensitive material in dumps, e.g. keys.                    |

## authpriv

Max level of messages logged in the private security/authorization messages facility.

```
vrouter running config# vrf <vrf> ike logging authpriv
```

## default

Default max log level.

```
vrouter running config# vrf <vrf> ike logging authpriv
vrouter running authpriv# default DEFAULT
```

| DEFAULT values | Description   |
|----------------|---|
| disable        | No log.   |
| 0              | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1              | Generic control flow with errors, a good default to see whats going on. |
| 2              | More detailed debugging control flow.                                   |
| 3              | Including RAW data dumps in hex.  |
| 4              | Also include sensitive material in dumps, e.g. keys.                    |

## Default value

disable

## asn1

Low-level encoding/decoding (ASN.1, X.509 etc.).

```
vrouter running config# vrf <vrf> ike logging authpriv
vrouter running authpriv# asn1 ASN1
```

| ASN1 values | Description   |
|-------------|---|
| disable     | No log.   |
| 0           | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1           | Generic control flow with errors, a good default to see whats going on. |
| 2           | More detailed debugging control flow.                                   |
| 3           | Including RAW data dumps in hex.  |
| 4           | Also include sensitive material in dumps, e.g. keys.                    |

## config

Configuration management and plugins.

```
vrouter running config# vrf <vrf> ike logging authpriv
vrouter running authpriv# config CONFIG
```

| CONFIG values | Description   |
|---------------|---|
| disable       | No log.   |
| 0             | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1             | Generic control flow with errors, a good default to see whats going on. |
| 2             | More detailed debugging control flow.                                   |
| 3             | Including RAW data dumps in hex.  |
| 4             | Also include sensitive material in dumps, e.g. keys.                    |

## child

CHILD\_SA/IPsec SA processing.

```
vrouter running config# vrf <vrf> ike logging authpriv
vrouter running authpriv# child CHILD
```

| CHILD values | Description   |
|--------------|---|
| disable      | No log.   |
| 0            | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1            | Generic control flow with errors, a good default to see whats going on. |
| 2            | More detailed debugging control flow.                                   |
| 3            | Including RAW data dumps in hex.  |
| 4            | Also include sensitive material in dumps, e.g. keys.                    |

## daemon

Main daemon setup/cleanup/signal handling.

```
vrouter running config# vrf <vrf> ike logging authpriv
vrouter running authpriv# daemon DAEMON
```

| DAEMON values | Description   |
|---------------|---|
| disable       | No log.   |
| 0             | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1             | Generic control flow with errors, a good default to see whats going on. |
| 2             | More detailed debugging control flow.                                   |
| 3             | Including RAW data dumps in hex.  |
| 4             | Also include sensitive material in dumps, e.g. keys.                    |

## encoding

Packet encoding/decoding encryption/decryption operations.

```
vrouter running config# vrf <vrf> ike logging authpriv
vrouter running authpriv# encoding ENCODING
```

| ENCODING values | Description   |
|-----------------|---|
| disable         | No log.   |
| 0               | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1               | Generic control flow with errors, a good default to see whats going on. |
| 2               | More detailed debugging control flow.                                   |
| 3               | Including RAW data dumps in hex.  |
| 4               | Also include sensitive material in dumps, e.g. keys.                    |

## ipsec

Libipsec library messages.

```
vrouter running config# vrf <vrf> ike logging authpriv
vrouter running authpriv# ipsec IPSEC
```

| IPSEC values | Description   |
|--------------|---|
| disable      | No log.   |
| 0            | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1            | Generic control flow with errors, a good default to see whats going on. |
| 2            | More detailed debugging control flow.                                   |
| 3            | Including RAW data dumps in hex.  |
| 4            | Also include sensitive material in dumps, e.g. keys.                    |

## ike

IKE\_SA/ISAKMP SA processing.

```
vrouter running config# vrf <vrf> ike logging authpriv
vrouter running authpriv# ike IKE
```

| IKE values | Description   |
|------------|---|
| disable    | No log.   |
| 0          | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1          | Generic control flow with errors, a good default to see whats going on. |
| 2          | More detailed debugging control flow.                                   |
| 3          | Including RAW data dumps in hex.  |
| 4          | Also include sensitive material in dumps, e.g. keys.                    |

## job

Jobs queuing/processing and thread pool management.

```
vrouter running config# vrf <vrf> ike logging authpriv
vrouter running authpriv# job JOB
```

| JOB values | Description   |
|------------|---|
| disable    | No log.   |
| 0          | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1          | Generic control flow with errors, a good default to see whats going on. |
| 2          | More detailed debugging control flow.                                   |
| 3          | Including RAW data dumps in hex.  |
| 4          | Also include sensitive material in dumps, e.g. keys.                    |

## kernel

IPsec/Networking kernel interface.

```
vrouter running config# vrf <vrf> ike logging authpriv
vrouter running authpriv# kernel KERNEL
```

| KERNEL values | Description   |
|---------------|---|
| disable       | No log.   |
| 0             | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1             | Generic control flow with errors, a good default to see whats going on. |
| 2             | More detailed debugging control flow.                                   |
| 3             | Including RAW data dumps in hex.  |
| 4             | Also include sensitive material in dumps, e.g. keys.                    |

## library

Libstrongwan library messages.

```
vrouter running config# vrf <vrf> ike logging authpriv
vrouter running authpriv# library LIBRARY
```

| LIBRARY values | Description   |
|----------------|---|
| disable        | No log.   |
| 0              | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1              | Generic control flow with errors, a good default to see whats going on. |
| 2              | More detailed debugging control flow.                                   |
| 3              | Including RAW data dumps in hex.  |
| 4              | Also include sensitive material in dumps, e.g. keys.                    |

## manager

IKE\_SA manager, handling synchronization for IKE\_SA access.

```
vrouter running config# vrf <vrf> ike logging authpriv
vrouter running authpriv# manager MANAGER
```

| MANAGER values | Description   |
|----------------|---|
| disable        | No log.   |
| 0              | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1              | Generic control flow with errors, a good default to see whats going on. |
| 2              | More detailed debugging control flow.                                   |
| 3              | Including RAW data dumps in hex.  |
| 4              | Also include sensitive material in dumps, e.g. keys.                    |

## network

IKE network communication.

```
vrouter running config# vrf <vrf> ike logging authpriv
vrouter running authpriv# network NETWORK
```

| NETWORK values | Description   |
|----------------|---|
| disable        | No log.   |
| 0              | Very basic auditing logs, (e.g. SA up/SA down).                         |
| 1              | Generic control flow with errors, a good default to see whats going on. |
| 2              | More detailed debugging control flow.                                   |
| 3              | Including RAW data dumps in hex.  |
| 4              | Also include sensitive material in dumps, e.g. keys.                    |

## global-options

Global ike options.

```
vrouter running config# vrf <vrf> ike global-options
```

## threads

Number of worker threads in IKE daemon.

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# threads <uint32>
```

## Default value

16

### acquire-timeout

Lifetime of SA acquire messages created when traffic matches a trap policy (seconds).

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# acquire-timeout <uint32>
```

#### Default value

30

### sa-table-size

Size of the IKE SA hash table.

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# sa-table-size <uint32>
```

#### Default value

1

### sa-table-segments

Number of locks to use for the IKE SA hash table.

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# sa-table-segments <uint32>
```

#### Default value

1

### install-routes

If true, install routes into a separate routing table for established IPsec tunnels.

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# install-routes true|false
```

#### Default value

false

## routing-table

Numerical routing table to install routes to.

```
vrouter running config# vrf <vrf> ike global-options  
vrouter running global-options# routing-table <uint32>
```

### Default value

220

## routing-table-prio

Priority of the routing table.

```
vrouter running config# vrf <vrf> ike global-options  
vrouter running global-options# routing-table-prio <uint32>
```

### Default value

220

## retransmit-tries

Number of times to retransmit a packet before giving up.

```
vrouter running config# vrf <vrf> ike global-options  
vrouter running global-options# retransmit-tries <0..100>
```

### Default value

5

## retransmit-timeout

Timeout in seconds before sending first retransmit.

```
vrouter running config# vrf <vrf> ike global-options  
vrouter running global-options# retransmit-timeout <0.000 .. 60.000>
```

### Default value

4.0

## retransmit-base

Base to use for calculating retransmit exponential back off.

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# retransmit-base <0.000 .. 10.000>
```

### Default value

1.8

## delete-rekeyed

Whether to immediately delete the old child SAs after an IKEv1 rekey. If false, old child SAs will be deleted after their hard lifetime, or on reception of a delete notification from the IKE peer.

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# delete-rekeyed true|false
```

### Default value

false

## delete-rekeyed-delay

Delay in seconds before deleting the old inbound child SAs after an IKEv2 rekey as initiator.

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# delete-rekeyed-delay DELETE-REKEYED-DELAY
```

| DELETE-REKEYED-DELAY values | Description                                   |
|-----------------------------|---|
| never                       | Keep the inbound child SA until its lifetime. |
| <uint32>                    | No description.                               |

### Default value

5

## make-before-break

During reauthentication, whether to recreate all new SAs before deleting the old ones. This implies to use overlapping IKE and child SAs, which must be supported by the IKE peer.

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# make-before-break true|false
```

### Default value

false

## interface-use

List of network interfaces that should be used. All other interfaces are ignored.

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# interface-use INTERFACE-USE
```

|               |                    |
|---------------|--------------------|
| INTERFACE-USE | An interface name. |
|---------------|--------------------|

## interface-ignore

List of network interfaces that should be ignored, if interfaces-use is specified this option has no effect.

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# interface-ignore INTERFACE-IGNORE
```

|                  |                    |
|------------------|--------------------|
| INTERFACE-IGNORE | An interface name. |
|------------------|--------------------|

## snmp

Enable or disable the IKE SNMP agent (default false).

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# snmp true|false
```

### Default value

false

### mobike-prefer-best-path

Dynamically update SAs with MOBIKE on routing changes using the cheapest path.

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# mobike-prefer-best-path true|false
```

#### Default value

false

### install-vip

Whether the virtual IP addresses should be installed.

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# install-vip true|false
```

#### Default value

true

### install-vip-on

The name of the interface on which virtual IP addresses should be installed. If not specified the addresses will be installed on the outbound interface.

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# install-vip-on INSTALL-VIP-ON
```

|                |                    |
|----------------|--------------------|
| INSTALL-VIP-ON | An interface name. |
|----------------|--------------------|

### retry-initiate-interval

Interval in seconds to use when retrying to initiate an IKE\_SA (e.g. if DNS resolution failed), 0 to disable retries.

```
vrouter running config# vrf <vrf> ike global-options
vrouter running global-options# retry-initiate-interval <uint8>
```

#### Default value

0

## dos-protection

Denial of Service protection using cookies and aggressiveness checks.

```
vrouter running config# vrf <vrf> ike global-options dos-protection
```

## cookie-threshold

Number of half-open IKE SAs that activate the cookie mechanism. 0 disables cookies.

```
vrouter running config# vrf <vrf> ike global-options dos-protection
vrouter running dos-protection# cookie-threshold COOKIE-THRESHOLD
```

| COOKIE-THRESHOLD values | Description                           |
|-------------------------|---------------------------------------|
| always                  | Always activate the cookie mechanism. |
| <uint32>                | No description.                       |

### Default value

10

## block-threshold

Maximum number of half-open IKE SAs for a single peer IP. 0 disables this limit.

```
vrouter running config# vrf <vrf> ike global-options dos-protection
vrouter running dos-protection# block-threshold <uint32>
```

### Default value

5

## init-limit-half-open

Refuse new connections if the current number of half open IKE SAs reaches this limit. 0 disables the limit.

```
vrouter running config# vrf <vrf> ike global-options dos-protection
vrouter running dos-protection# init-limit-half-open <uint32>
```

### Default value

0

## sp-hash-ipv4

Thresholds for hashing IPv4 Security Policies in IPsec stack.

```
vrouter running config# vrf <vrf> ike global-options  
vrouter running global-options# sp-hash-ipv4 local <uint8> remote <uint8>
```

### local

Number of sp local address bits to include in hash key.

```
local <uint8>
```

### Default value

32

### remote

Number of sp remote address bits to include in hash key.

```
remote <uint8>
```

### Default value

32

## sp-hash-ipv6

Thresholds for hashing IPv6 Security Policies in IPsec stack.

```
vrouter running config# vrf <vrf> ike global-options  
vrouter running global-options# sp-hash-ipv6 local <uint8> remote <uint8>
```

### local

Number of sp local address bits to include in hash key.

```
local <uint8>
```

### Default value

128

## remote

Number of sp remote address bits to include in hash key.

```
remote <uint8>
```

### Default value

128

## ha

IKE High Availability parameters.

```
vrouter running config# vrf <vrf> ike ha
```

## enabled

Enable or disable IKE High Availability.

```
vrouter running config# vrf <vrf> ike ha  
vrouter running ha# enabled true|false
```

### Default value

true

## listen-ha-group (mandatory)

The HA group to be monitored. If the state of this group changes, it will trigger a failover of the IKE service to/from another IKE HA node.

```
vrouter running config# vrf <vrf> ike ha  
vrouter running ha# listen-ha-group <string>
```

## node-id (mandatory)

Local identifier in the IKE HA Cluster.

```
vrouter running config# vrf <vrf> ike ha  
vrouter running ha# node-id <int8>
```

**interface (mandatory)**

Interface on which to perform HA peer discovery.

```
vrouter running config# vrf <vrf> ike ha
vrouter running ha# interface INTERFACE
```

|           |                    |
|-----------|--------------------|
| INTERFACE | An interface name. |
|-----------|--------------------|

**local-address (mandatory)**

Local IP address to communicate with the HA peer.

```
vrouter running config# vrf <vrf> ike ha
vrouter running ha# local-address LOCAL-ADDRESS
```

| LOCAL-ADDRESS values | Description      |
|----------------------|------------------|
| <ipv4-address>       | An IPv4 address. |
| <ipv6-address>       | An IPv6 address. |

**remote-address (mandatory)**

Remote IP address to communicate with the HA peer.

```
vrouter running config# vrf <vrf> ike ha
vrouter running ha# remote-address REMOTE-ADDRESS
```

| REMOTE-ADDRESS values | Description      |
|-----------------------|------------------|
| <ipv4-address>        | An IPv4 address. |
| <ipv6-address>        | An IPv6 address. |

**seqnum-sync**

SA sequence number synchronization.

```
vrouter running config# vrf <vrf> ike ha seqnum-sync
```

## oseq-shift

SA output sequence number advance on backup node.

```
vrouter running config# vrf <vrf> ike ha seqnum-sync
vrouter running seqnum-sync# oseq-shift <uint64>
```

### Default value

65536

## sync-period-time

SA sequence number synchronization period in time. State is always printed in seconds.

```
vrouter running config# vrf <vrf> ike ha seqnum-sync
vrouter running seqnum-sync# sync-period-time SYNC-PERIOD-TIME
```

|                  |   |
|------------------|---|
| SYNC-PERIOD-TIME | IKE duration, with optional unit (s m h d). |
|------------------|---|

### Default value

10s

## sync-period-packets

SA sequence number synchronization period in packets.

```
vrouter running config# vrf <vrf> ike ha seqnum-sync
vrouter running seqnum-sync# sync-period-packets <uint32>
```

### Default value

2

## pool

List of virtual address pools synchronized via HA.

```
vrouter running config# vrf <vrf> ike ha pool <pool>
```

|        |                       |
|--------|-----------------------|
| <pool> | IKE object name type. |
|--------|-----------------------|

### address (mandatory)

Virtual addresses in the pool.

```
vrouter running config# vrf <vrf> ike ha pool <pool>
vrouter running pool <pool># address ADDRESS
```

| ADDRESS values | Description                            |
|----------------|--|
| <ipv4-prefix>  | An IPv4 prefix: address and CIDR mask. |
| <ipv6-prefix>  | An IPv6 prefix: address and CIDR mask. |

### ike-policy-template (config only)

List of IKE VPN policies.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template>
```

|                       |                       |
|-----------------------|-----------------------|
| <ike-policy-template> | IKE object name type. |
|-----------------------|-----------------------|

### local-auth-method (config only)

Local IKE authentication method.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template>
vrouter running ike-policy-template <ike-policy-template># local-auth-method LOCAL-
AUTH-METHOD
```

| LOCAL-AUTH-METHOD values | Description  |
|--------------------------|--|
| pre-shared-key           | Pre-shared key.  |
| certificate              | Public key signature with X509 Certificates.   |
| eap-md5                  | Extensible Authentication Protocol - MD5-Challenge.  |
| eap-mschapv2             | Extensible Authentication Protocol - Microsoft Challenge-Handshake Authentication Protocol v2. |

### Default value

pre-shared-key

### remote-auth-method (config only)

Remote IKE authentication method.

```
vrouters running config# vrf <vrf> ike ike-policy-template <ike-policy-template>
vrouters running ike-policy-template <ike-policy-template># remote-auth-method REMOTE-
AUTH-METHOD
```

| REMOTE-AUTH-METHOD values | Description  |
|---------------------------|--|
| pre-shared-key            | Pre-shared key.  |
| certificate               | Public key signature with X509 Certificates.   |
| eap-md5                   | Extensible Authentication Protocol - MD5-Challenge.  |
| eap-mschapv2              | Extensible Authentication Protocol - Microsoft Challenge-Handshake Authentication Protocol v2. |
| eap-radius                | Extensible Authentication Protocol delegated to a RADIUS server.                               |

#### Default value

pre-shared-key

### keying-tries (config only)

Number of times we should try to initiate an IKE connection if the responder does not answer (after a full sequence of retransmissions). A value of 0 initiates a new sequence forever, until the connection establishes or fails with a permanent error.

```
vrouters running config# vrf <vrf> ike ike-policy-template <ike-policy-template>
vrouters running ike-policy-template <ike-policy-template># keying-tries <uint32>
```

#### Default value

1

### unique-sa (config only)

Connection uniqueness policy to enforce, to avoid multiple connections from the same user ID.

```
vrouters running config# vrf <vrf> ike ike-policy-template <ike-policy-template>
vrouters running ike-policy-template <ike-policy-template># unique-sa UNIQUE-SA
```

| UNIQUE-SA values | Description   |
|------------------|---|
| no               | Do not enforce IKE SA uniqueness, except if a peer included INITIAL_CONTACT notify.                                       |
| never            | Never enforce IKE SA uniqueness, even if a peer included INITIAL_CONTACT notify. Never send INITIAL_CONTACT as initiator. |
| keep             | Reject new connection attempts from same user.  |
| replace          | Delete any existing connection if a new one for the same user gets established.   |

**Default value**

no

**reauth-time (config only)**

Time to schedule IKE reauthentication.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template>
vrouter running ike-policy-template <ike-policy-template># reauth-time REAUTH-TIME
```

|             |   |
|-------------|---|
| REAUTH-TIME | IKE duration, with optional unit (s m h d). |
|-------------|---|

**Default value**

0s

**rekey-time (config only)**

Time to schedule IKE rekeying.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template>
vrouter running ike-policy-template <ike-policy-template># rekey-time REKEY-TIME
```

|            |   |
|------------|---|
| REKEY-TIME | IKE duration, with optional unit (s m h d). |
|------------|---|

**Default value**

4h

### dpd-delay (config only)

Interval to check the liveness of a peer.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template>  
vrouter running ike-policy-template <ike-policy-template># dpd-delay DPD-DELAY
```

|           |   |
|-----------|---|
| DPD-DELAY | IKE duration, with optional unit (s m h d). |
|-----------|---|

#### Default value

0s

### aggressive (config only)

Enable or disable Aggressive Mode instead of Main Mode in IKEv1.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template>  
vrouter running ike-policy-template <ike-policy-template># aggressive true|false
```

#### Default value

false

### udp-encap (config only)

If true, enforce UDP encapsulation of ESP packets.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template>  
vrouter running ike-policy-template <ike-policy-template># udp-encap true|false
```

#### Default value

false

### mobike (config only)

If true, enable MOBIKE (IKEv2 Mobility and Multihoming Protocol).

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template>  
vrouter running ike-policy-template <ike-policy-template># mobike true|false
```

#### Default value

false

### ike-proposal (config only)

List of IKE phase 1 proposals.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template> ike-
↳proposal <uint8>
```

|         |   |
|---------|---|
| <uint8> | Index in the list of IKE phase 1 proposals. |
|---------|---|

### enc-alg (config only)

List of encryption algorithms for IKE SAs.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template> ike-
↳proposal <uint8>
vrouter running ike-proposal <uint8># enc-alg ENC-ALG
```

| ENC-ALG values  | Description                |
|-----------------|----------------------------|
| aes128-cbc      | AES-CBC, 128 bit key.      |
| aes192-cbc      | AES-CBC, 192 bit key.      |
| aes256-cbc      | AES-CBC, 256 bit key.      |
| des-cbc         | DES-CBC, 56 bit key.       |
| 3des-cbc        | 3DES-CBC, 168 bit key.     |
| aes128-ctr      | AES-CTR, 128 bit key.      |
| aes192-ctr      | AES-CTR, 192 bit key.      |
| aes256-ctr      | AES-CTR, 256 bit key.      |
| cast-cbc        | CAST-CBC, 128 bit key.     |
| blowfish128-cbc | Blowfish-CBC, 128 bit key. |
| blowfish192-cbc | Blowfish-CBC, 192 bit key. |
| blowfish256-cbc | Blowfish-CBC, 256 bit key. |
| camellia128-cbc | Camellia-CBC, 128 bit key. |
| camellia192-cbc | Camellia-CBC, 192 bit key. |
| camellia256-cbc | Camellia-CBC, 256 bit key. |
| camellia128-ctr | Camellia-CTR, 128 bit key. |
| camellia192-ctr | Camellia-CTR, 192 bit key. |
| camellia256-ctr | Camellia-CTR, 256 bit key. |

### auth-alg (config only)

List of auth algorithms for IKE SAs.

```
vrouters running config# vrf <vrf> ike ike-policy-template <ike-policy-template> ike-
↳proposal <uint8>
vrouters running ike-proposal <uint8># auth-alg AUTH-ALG
```

| AUTH-ALG values | Description      |
|-----------------|------------------|
| hmac-md5        | HMAC-MD5-96.     |
| hmac-sha1       | HMAC-SHA1-96.    |
| hmac-sha256     | HMAC-SHA256-128. |
| hmac-sha384     | HMAC-SHA384-192. |
| hmac-sha512     | HMAC-SHA512-256. |
| aes-xcbc        | AES-XCBC-96.     |

### aead-alg (config only)

List of combined-mode (AEAD) algorithms for IKE SAs.

```
vrouters running config# vrf <vrf> ike ike-policy-template <ike-policy-template> ike-
↳proposal <uint8>
vrouters running ike-proposal <uint8># aead-alg AEAD-ALG
```

| AEAD-ALG values    | Description                            |
|--------------------|--|
| aes128-gcm-64      | AES-GCM, 128 bit key, 64 bit ICV.      |
| aes192-gcm-64      | AES-GCM, 192 bit key, 64 bit ICV.      |
| aes256-gcm-64      | AES-GCM, 256 bit key, 64 bit ICV.      |
| aes128-gcm-96      | AES-GCM, 128 bit key, 96 bit ICV.      |
| aes192-gcm-96      | AES-GCM, 192 bit key, 96 bit ICV.      |
| aes256-gcm-96      | AES-GCM, 256 bit key, 96 bit ICV.      |
| aes128-gcm-128     | AES-GCM, 128 bit key, 128 bit ICV.     |
| aes192-gcm-128     | AES-GCM, 192 bit key, 128 bit ICV.     |
| aes256-gcm-128     | AES-GCM, 256 bit key, 128 bit ICV.     |
| aes128-ccm-64      | AES-CCM, 128 bit key, 64 bit ICV.      |
| aes192-ccm-64      | AES-CCM, 192 bit key, 64 bit ICV.      |
| aes256-ccm-64      | AES-CCM, 256 bit key, 64 bit ICV.      |
| aes128-ccm-96      | AES-CCM, 128 bit key, 96 bit ICV.      |
| aes192-ccm-96      | AES-CCM, 192 bit key, 96 bit ICV.      |
| aes256-ccm-96      | AES-CCM, 256 bit key, 96 bit ICV.      |
| aes128-ccm-128     | AES-CCM, 128 bit key, 128 bit ICV.     |
| aes192-ccm-128     | AES-CCM, 192 bit key, 128 bit ICV.     |
| aes256-ccm-128     | AES-CCM, 256 bit key, 128 bit ICV.     |
| camellia128-ccm-64 | Camellia-CCM, 128 bit key, 64 bit ICV. |
| camellia192-ccm-64 | Camellia-CCM, 192 bit key, 64 bit ICV. |
| camellia256-ccm-64 | Camellia-CCM, 256 bit key, 64 bit ICV. |
| camellia128-ccm-96 | Camellia-CCM, 128 bit key, 96 bit ICV. |
| camellia192-ccm-96 | Camellia-CCM, 192 bit key, 96 bit ICV. |
| camellia256-ccm-96 | Camellia-CCM, 256 bit key, 96 bit ICV. |

### prf-alg (config only)

List of pseudo-random algorithms for IKE SAs.

```
vrouters running config# vrf <vrf> ike ike-policy-template <ike-policy-template> ike-
  ↪proposal <uint8>
vrouters running ike-proposal <uint8># prf-alg PRF-ALG
```

| PRF-ALG values | Description       |
|----------------|-------------------|
| hmac-md5       | PRF-HMAC-MD5.     |
| hmac-sha1      | PRF-HMAC-SHA1.    |
| aes-xcbc       | AES-XCBC-PRF-128. |
| aes-cmac       | AES-CMAC-PRF-128. |
| hmac-sha256    | PRF-HMAC-SHA-256. |
| hmac-sha384    | PRF-HMAC-SHA-384. |
| hmac-sha512    | PRF-HMAC-SHA-512. |

## dh-group (config only)

List of Diffie Hellman groups for key exchange.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template> ike-
↳proposal <uint8>
vrouter running ike-proposal <uint8># dh-group DH-GROUP
```

| DH-GROUP values | Description   |
|-----------------|---|
| modp768         | Modulo Prime 768 bits (group 1).                      |
| modp1024        | Modulo Prime 1024 bits (group 2).                     |
| modp1536        | Modulo Prime 1536 bits (group 5).                     |
| modp2048        | Modulo Prime 2048 bits (group 14).                    |
| modp3072        | Modulo Prime 3072 bits (group 15).                    |
| modp4096        | Modulo Prime 4096 bits (group 16).                    |
| modp6144        | Modulo Prime 6144 bits (group 17).                    |
| modp8192        | Modulo Prime 8192 bits (group 18).                    |
| modp1024s160    | Modulo Prime 1024 bits, Subgroup 160 bits (group 22). |
| modp1024s224    | Modulo Prime 1024 bits, Subgroup 224 bits (group 23). |
| modp1024s256    | Modulo Prime 1024 bits, Subgroup 256 bits (group 24). |
| ecp192          | Elliptic Curve 192 bits (group 25).                   |
| ecp224          | Elliptic Curve 224 bits (group 26).                   |
| ecp256          | Elliptic Curve 256 bits (group 19).                   |
| ecp384          | Elliptic Curve 384 bits (group 20).                   |
| ecp521          | Elliptic Curve 521 bits (group 21).                   |
| ecp224bp        | Brainpool Elliptic Curve 224 bits (group 27).         |
| ecp256bp        | Brainpool Elliptic Curve 256 bits (group 28).         |
| ecp384bp        | Brainpool Elliptic Curve 384 bits (group 29).         |
| ecp512bp        | Brainpool Elliptic Curve 512 bits (group 30).         |

## ipsec-policy-template (config only)

List of IPsec VPN policies.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
```

|                         |                       |
|-------------------------|-----------------------|
| <ipsec-policy-template> | IKE object name type. |
|-------------------------|-----------------------|

**start-action (config only)**

Action to perform for this CHILD\_SA on DPD timeout.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
vrouter running ipsec-policy-template <ipsec-policy-template># start-action START-
↳ACTION
```

| START-ACTION values | Description   |
|---------------------|---|
| none                | Load the connection only, can be used as a responder configuration.                             |
| trap                | Install a trap policy, which triggers the tunnel as soon as matching traffic has been detected. |
| start               | Initiate the connection actively.   |

**Default value**

trap

**close-action (config only)**

Action to perform when a CHILD\_SA gets closed by a peer.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
vrouter running ipsec-policy-template <ipsec-policy-template># close-action CLOSE-
↳ACTION
```

| CLOSE-ACTION values | Description  |
|---------------------|--|
| none                | Close the Child SA and take no further action.                                       |
| trap                | Install a trap policy matching traffic and try to re-negotiate the tunnel on-demand. |
| start               | Try to immediately re-create the CHILD_SA.   |

**Default value**

trap

**dpd-action (config only)**

Action to perform for a CHILD\_SA on DPD timeout.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
vrouter running ipsec-policy-template <ipsec-policy-template># dpd-action DPD-ACTION
```

| DPD-ACTION values | Description   |
|-------------------|---|
| clear             | Close the Child SA and take no further action.  |
| trap              | Install a trap policy, which will catch matching traffic and tries to re-negotiate the tunnel on-demand action. |
| restart           | Immediately try to re-negotiate the CHILD_SA under a fresh IKE_SA.  |

**Default value**

restart

**replay-window (config only)**

Replay window size. 0 disables IPsec replay protection.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
vrouter running ipsec-policy-template <ipsec-policy-template># replay-window <uint16>
```

**Default value**

32

**rekey-time (config only)**

Time before initiating CHILD\_SA rekeying.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
vrouter running ipsec-policy-template <ipsec-policy-template># rekey-time REKEY-TIME
```

|            |   |
|------------|---|
| REKEY-TIME | IKE duration, with optional unit (s m h d). |
|------------|---|

**Default value**

1h

### life-time (config only)

Maximum lifetime before CHILD\_SA gets closed (default rekey-time + 10%).

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>  
vrouter running ipsec-policy-template <ipsec-policy-template># life-time LIFE-TIME
```

|           |   |
|-----------|---|
| LIFE-TIME | IKE duration, with optional unit (s m h d). |
|-----------|---|

### rand-time (config only)

Time range from which to choose a random value to subtract from rekey\_time (default life\_time - rekey\_time).

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>  
vrouter running ipsec-policy-template <ipsec-policy-template># rand-time RAND-TIME
```

|           |   |
|-----------|---|
| RAND-TIME | IKE duration, with optional unit (s m h d). |
|-----------|---|

### rekey-bytes (config only)

Number of bytes processed before initiating CHILD\_SA rekeying.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>  
vrouter running ipsec-policy-template <ipsec-policy-template># rekey-bytes <uint64>
```

#### Default value

0

### life-bytes (config only)

Maximum bytes processed before CHILD\_SA gets closed (default rekey-bytes + 10%).

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>  
vrouter running ipsec-policy-template <ipsec-policy-template># life-bytes <uint64>
```

### rand-bytes (config only)

Byte range from which to choose a random value to subtract from rekey\_bytes (default life\_bytes - rekey\_bytes).

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
vrouter running ipsec-policy-template <ipsec-policy-template># rand-bytes <uint64>
```

### rekey-packets (config only)

Number of packets processed before initiating CHILD\_SA rekeying.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
vrouter running ipsec-policy-template <ipsec-policy-template># rekey-packets <uint64>
```

#### Default value

0

### life-packets (config only)

Maximum packets processed before CHILD\_SA gets closed (default rekey\_bytes + 10%).

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
vrouter running ipsec-policy-template <ipsec-policy-template># life-packets <uint64>
```

### rand-packets (config only)

Packet range from which to choose a random value to subtract from rekey\_packets (default life\_bytes - rekey\_bytes).

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
vrouter running ipsec-policy-template <ipsec-policy-template># rand-packets <uint64>
```

### encap-copy-dscp (config only)

Whether to copy DSCP from inner to outer IP header at IPsec encapsulation.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
vrouter running ipsec-policy-template <ipsec-policy-template># encap-copy-dscp_
↪ true|false
```

#### Default value

true

**decap-copy-dscp (config only)**

Whether to copy DSCP from outer to inner IP header at IPsec decapsulation.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
vrouter running ipsec-policy-template <ipsec-policy-template># decap-copy-dscp
↳ true|false
```

**Default value**

false

**encap-copy-df (config only)**

Whether to copy the Don't Fragment bit from outer to inner IP header at IPsec encapsulation.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
vrouter running ipsec-policy-template <ipsec-policy-template># encap-copy-df true|false
```

**Default value**

true

**esp-proposal (config only)**

List of ESP proposals.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
↳ esp-proposal <uint8>
```

|         |                                 |
|---------|---------------------------------|
| <uint8> | Index in list of ESP proposals. |
|---------|---------------------------------|

**enc-alg (config only)**

List of encryption algorithms for IPsec SAs.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
↳ esp-proposal <uint8>
vrouter running esp-proposal <uint8># enc-alg ENC-ALG
```

| ENC-ALG values | Description            |
|----------------|------------------------|
| null           | NULL.                  |
| aes128-cbc     | AES-CBC, 128 bit key.  |
| aes192-cbc     | AES-CBC, 192 bit key.  |
| aes256-cbc     | AES-CBC, 256 bit key.  |
| des-cbc        | DES-CBC, 56 bit key.   |
| 3des-cbc       | 3DES-CBC, 168 bit key. |

### auth-alg (config only)

List of auth algorithms for IPsec SAs.

```
vrouters running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
↳ esp-proposal <uint8>
vrouters running esp-proposal <uint8># auth-alg AUTH-ALG
```

| AUTH-ALG values | Description      |
|-----------------|------------------|
| none            | NONE.            |
| hmac-md5        | HMAC-MD5-96.     |
| hmac-sha1       | HMAC-SHA1-96.    |
| hmac-sha256     | HMAC-SHA256-128. |
| hmac-sha384     | HMAC-SHA384-192. |
| hmac-sha512     | HMAC-SHA512-256. |
| aes-xcbc        | AES-XCBC-96.     |

### aead-alg (config only)

List of combined-mode (AEAD) algorithms for IPsec SAs.

```
vrouters running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
↳ esp-proposal <uint8>
vrouters running esp-proposal <uint8># aead-alg AEAD-ALG
```

| AEAD-ALG values | Description                         |
|-----------------|-------------------------------------|
| aes128-gcm-128  | AES-GCM, 128 bit key, 128 bit ICV.  |
| aes192-gcm-128  | AES-GCM, 192 bit key, 128 bit ICV.  |
| aes256-gcm-128  | AES-GCM, 256 bit key, 128 bit ICV.  |
| aes128-gmac     | AES-GMAC, 128 bit key, 128 bit ICV. |
| aes192-gmac     | AES-GMAC, 192 bit key, 128 bit ICV. |
| aes256-gmac     | AES-GMAC, 256 bit key, 128 bit ICV. |

## dh-group (config only)

List of Diffie Hellman groups for Perfect Forward Secrecy.

```
vrouters running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
↳ esp-proposal <uint8>
vrouters running esp-proposal <uint8># dh-group DH-GROUP
```

| DH-GROUP values | Description   |
|-----------------|---|
| modp768         | Modulo Prime 768 bits (group 1).                      |
| modp1024        | Modulo Prime 1024 bits (group 2).                     |
| modp1536        | Modulo Prime 1536 bits (group 5).                     |
| modp2048        | Modulo Prime 2048 bits (group 14).                    |
| modp3072        | Modulo Prime 3072 bits (group 15).                    |
| modp4096        | Modulo Prime 4096 bits (group 16).                    |
| modp6144        | Modulo Prime 6144 bits (group 17).                    |
| modp8192        | Modulo Prime 8192 bits (group 18).                    |
| modp1024s160    | Modulo Prime 1024 bits, Subgroup 160 bits (group 22). |
| modp1024s224    | Modulo Prime 1024 bits, Subgroup 224 bits (group 23). |
| modp1024s256    | Modulo Prime 1024 bits, Subgroup 256 bits (group 24). |
| ecp192          | Elliptic Curve 192 bits (group 25).                   |
| ecp224          | Elliptic Curve 224 bits (group 26).                   |
| ecp256          | Elliptic Curve 256 bits (group 19).                   |
| ecp384          | Elliptic Curve 384 bits (group 20).                   |
| ecp521          | Elliptic Curve 521 bits (group 21).                   |
| ecp224bp        | Brainpool Elliptic Curve 224 bits (group 27).         |
| ecp256bp        | Brainpool Elliptic Curve 256 bits (group 28).         |
| ecp384bp        | Brainpool Elliptic Curve 384 bits (group 29).         |
| ecp512bp        | Brainpool Elliptic Curve 512 bits (group 30).         |

## esn (config only)

List of Extended Sequence Number modes.

```
vrouters running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
↳ esp-proposal <uint8>
vrouters running esp-proposal <uint8># esn true|false
```

### ah-proposal (config only)

List of AH proposals.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template> ah-
↳proposal <uint8>
```

|         |                                |
|---------|--------------------------------|
| <uint8> | Index in list of AH proposals. |
|---------|--------------------------------|

### auth-alg (config only)

List of auth algorithms for IPsec SAs.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template> ah-
↳proposal <uint8>
vrouter running ah-proposal <uint8># auth-alg AUTH-ALG
```

| AUTH-ALG values | Description      |
|-----------------|------------------|
| hmac-md5        | HMAC-MD5-96.     |
| hmac-sha1       | HMAC-SHA1-96.    |
| hmac-sha256     | HMAC-SHA256-128. |
| hmac-sha384     | HMAC-SHA384-192. |
| hmac-sha512     | HMAC-SHA512-256. |
| aes-xcbc        | AES-XCBC-96.     |

### dh-group (config only)

List of Diffie Hellman groups for Perfect Forward Secrecy.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template> ah-
↳proposal <uint8>
vrouter running ah-proposal <uint8># dh-group DH-GROUP
```

| DH-GROUP values | Description   |
|-----------------|---|
| modp768         | Modulo Prime 768 bits (group 1).                      |
| modp1024        | Modulo Prime 1024 bits (group 2).                     |
| modp1536        | Modulo Prime 1536 bits (group 5).                     |
| modp2048        | Modulo Prime 2048 bits (group 14).                    |
| modp3072        | Modulo Prime 3072 bits (group 15).                    |
| modp4096        | Modulo Prime 4096 bits (group 16).                    |
| modp6144        | Modulo Prime 6144 bits (group 17).                    |
| modp8192        | Modulo Prime 8192 bits (group 18).                    |
| modp1024s160    | Modulo Prime 1024 bits, Subgroup 160 bits (group 22). |
| modp1024s224    | Modulo Prime 1024 bits, Subgroup 224 bits (group 23). |
| modp1024s256    | Modulo Prime 1024 bits, Subgroup 256 bits (group 24). |
| ecp192          | Elliptic Curve 192 bits (group 25).                   |
| ecp224          | Elliptic Curve 224 bits (group 26).                   |
| ecp256          | Elliptic Curve 256 bits (group 19).                   |
| ecp384          | Elliptic Curve 384 bits (group 20).                   |
| ecp521          | Elliptic Curve 521 bits (group 21).                   |
| ecp224bp        | Brainpool Elliptic Curve 224 bits (group 27).         |
| ecp256bp        | Brainpool Elliptic Curve 256 bits (group 28).         |
| ecp384bp        | Brainpool Elliptic Curve 384 bits (group 29).         |
| ecp512bp        | Brainpool Elliptic Curve 512 bits (group 30).         |

### esn (config only)

List of Extended Sequence Number modes.

```
vrouters running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template> ah-
↳proposal <uint8>
vrouters running ah-proposal <uint8># esn true|false
```

### vpn

List of IKE Virtual Private Networks.

```
vrouters running config# vrf <vrf> ike vpn <vpn>
```

|       |                       |
|-------|-----------------------|
| <vpn> | IKE object name type. |
|-------|-----------------------|

## description

Description of the VPN.

```
vrouter running config# vrf <vrf> ike vpn <vpn>  
vrouter running vpn <vpn># description <string>
```

## version

IKE version. 0 accepts both IKEv1 and IKEv2 as responder, and initiates the connection actively with IKEv2.

```
vrouter running config# vrf <vrf> ike vpn <vpn>  
vrouter running vpn <vpn># version <uint8>
```

## Default value

2

## local-address

List of IKE local peer addresses.

```
vrouter running config# vrf <vrf> ike vpn <vpn>  
vrouter running vpn <vpn># local-address LOCAL-ADDRESS
```

| LOCAL values   | Description  |
|----------------|--|
| <domain-name>  | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <ipv4-address> | An IPv4 address.   |
| <ipv6-address> | An IPv6 address.   |
| <ipv4-prefix>  | An IPv4 prefix: address and CIDR mask.   |
| <ipv6-prefix>  | An IPv6 prefix: address and CIDR mask.   |
| <ipv4-range>   | An IPv4 address range, in the form addr4-addr4.  |
| <ipv6-range>   | An IPv6 address range, in the form addr6-addr6.  |

### remote-address

List of IKE remote peer addresses.

```
vrouter running config# vrf <vrf> ike vpn <vpn>
vrouter running vpn <vpn># remote-address REMOTE-ADDRESS
```

| REMOTE         | DESCRIPTION  |
|----------------|--|
| val-<br>ues    |  |
| <domain-name>  | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <ipv4-address> | An IPv4 address.   |
| <ipv6-address> | An IPv6 address.   |
| <ipv4-prefix>  | An IPv4 prefix: address and CIDR mask.   |
| <ipv6-prefix>  | An IPv6 prefix: address and CIDR mask.   |
| <ipv4-range>   | An IPv4 address range, in the form addr4-addr4.  |
| <ipv6-range>   | An IPv6 address range, in the form addr6-addr6.  |

### local-id

Local IKE identifier (IP address, fqdn, user-fqdn, ASN.1 Distinguished Name) (Default psk: IP address, certificates: SubjectName).

```
vrouter running config# vrf <vrf> ike vpn <vpn>
vrouter running vpn <vpn># local-id LOCAL-ID
```

| LOCAL Description | Values   |
|-------------------|--|
| <ike-id>          | An IPv4 address.   |
| <ike-id>          | An IPv6 address.   |
| <ike-id>          | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <ike-id>          | IKE ID (IP address, fqdn, e-mail address or distinguished name).   |
| <ike-id>          | IKE ID (IP address, fqdn, e-mail address or distinguished name).   |

### remote-id

Remote IKE identifier (IP address, fqdn, user-fqdn, ASN.1 Distinguished Name) (Default psk: IP address, certificates: SubjectName).

```
vrouter running config# vrf <vrf> ike vpn <vpn>
vrouter running vpn <vpn># remote-id REMOTE-ID
```

| REMOTE values | Description  |
|---------------|--|
| <ike-id>      | An IPv4 address.   |
| <ike-id>      | An IPv6 address.   |
| <ike-id>      | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <ike-id>      | IKE ID (IP address, fqdn, e-mail address or distinguished name).   |
| <ike-id>      | IKE ID (IP address, fqdn, e-mail address or distinguished name).   |

### local-eap-id

Local EAP identifier (Default = local-id).

```
vrouter running config# vrf <vrf> ike vpn <vpn>
vrouter running vpn <vpn># local-eap-id LOCAL-EAP-ID
```

|              |         |
|--------------|---------|
| LOCAL-EAP-ID | EAP ID. |
|--------------|---------|

### remote-eap-id

Remote EAP identifier (Default = remote-id).

```
vrouter running config# vrf <vrf> ike vpn <vpn>
vrouter running vpn <vpn># remote-eap-id REMOTE-EAP-ID
```

|               |         |
|---------------|---------|
| REMOTE-EAP-ID | EAP ID. |
|---------------|---------|

## certificate

List of certificates to use for authentication of the local peer.

```
vrouter running config# vrf <vrf> ike vpn <vpn>
vrouter running vpn <vpn># certificate <leafref>
```

## remote-ca-certificate

List of certificate authority certificates to accept for authentication of the remote peer.

```
vrouter running config# vrf <vrf> ike vpn <vpn>
vrouter running vpn <vpn># remote-ca-certificate <leafref>
```

## vip-request

List of virtual IP addresses to request (0.0.0.0 for any IPv4 address, :: for any IPv6 address).

```
vrouter running config# vrf <vrf> ike vpn <vpn>
vrouter running vpn <vpn># vip-request VIP-REQUEST
```

| VIP-REQUEST values | Description      |
|--------------------|------------------|
| <ipv4-address>     | An IPv4 address. |
| <ipv6-address>     | An IPv6 address. |

## vip-pool

List of virtual IP pools, to assign a virtual IP to an IKE peer.

```
vrouter running config# vrf <vrf> ike vpn <vpn>
vrouter running vpn <vpn># vip-pool <leafref>
```

## dynamic-svti (config only)

Dynamic SVTI interfaces creation.

```
vrouter running config# vrf <vrf> ike vpn <vpn> dynamic-svti
```

**svti-template (config only) (mandatory)**

Dynamic SVTI template.

```
vrouter running config# vrf <vrf> ike vpn <vpn> dynamic-svti
vrouter running dynamic-svti# svti-template <leafref>
```

**vrf (config only)**

Dynamic SVTI template vrf.

```
vrouter running config# vrf <vrf> ike vpn <vpn> dynamic-svti
vrouter running dynamic-svti# vrf VRF
```

| VRF values | Description   |
|------------|---------------|
| main       | The main vrf. |
| <string>   | The vrf name. |

**ike-policy**

IKE policy configuration.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy
```

**template (config only) (mandatory)**

Template from which this IKE policy derives.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy
vrouter running ike-policy# template <leafref>
```

**local-auth-method**

Local IKE authentication method.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy
vrouter running ike-policy# local-auth-method LOCAL-AUTH-METHOD
```

| LOCAL-AUTH-METHOD values | Description  |
|--------------------------|--|
| pre-shared-key           | Pre-shared key.  |
| certificate              | Public key signature with X509 Certificates.   |
| eap-md5                  | Extensible Authentication Protocol - MD5-Challenge.  |
| eap-mschapv2             | Extensible Authentication Protocol - Microsoft Challenge-Handshake Authentication Protocol v2. |

### remote-auth-method

Remote IKE authentication method.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy
vrouter running ike-policy# remote-auth-method REMOTE-AUTH-METHOD
```

| REMOTE-AUTH-METHOD values | Description  |
|---------------------------|--|
| pre-shared-key            | Pre-shared key.  |
| certificate               | Public key signature with X509 Certificates.   |
| eap-md5                   | Extensible Authentication Protocol - MD5-Challenge.  |
| eap-mschapv2              | Extensible Authentication Protocol - Microsoft Challenge-Handshake Authentication Protocol v2. |
| eap-radius                | Extensible Authentication Protocol delegated to a RADIUS server.                               |

### keying-tries

Number of times we should try to initiate an IKE connection if the responder does not answer (after a full sequence of retransmissions). A value of 0 initiates a new sequence forever, until the connection establishes or fails with a permanent error.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy
vrouter running ike-policy# keying-tries <uint32>
```

## unique-sa

Connection uniqueness policy to enforce, to avoid multiple connections from the same user ID.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy
vrouter running ike-policy# unique-sa UNIQUE-SA
```

| UNIQUE-SA values | Description   |
|------------------|---|
| no               | Do not enforce IKE SA uniqueness, except if a peer included INITIAL_CONTACT notify.                                       |
| never            | Never enforce IKE SA uniqueness, even if a peer included INITIAL_CONTACT notify. Never send INITIAL_CONTACT as initiator. |
| keep             | Reject new connection attempts from same user.  |
| replace          | Delete any existing connection if a new one for the same user gets established.   |

## reauth-time

Time to schedule IKE reauthentication.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy
vrouter running ike-policy# reauth-time REAUTH-TIME
```

|             |   |
|-------------|---|
| REAUTH-TIME | IKE duration, with optional unit (s m h d). |
|-------------|---|

## rekey-time

Time to schedule IKE rekeying.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy
vrouter running ike-policy# rekey-time REKEY-TIME
```

|            |   |
|------------|---|
| REKEY-TIME | IKE duration, with optional unit (s m h d). |
|------------|---|

## dpd-delay

Interval to check the liveness of a peer.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy  
vrouter running ike-policy# dpd-delay DPD-DELAY
```

|           |   |
|-----------|---|
| DPD-DELAY | IKE duration, with optional unit (s m h d). |
|-----------|---|

## aggressive

Enable or disable Aggressive Mode instead of Main Mode in IKEv1.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy  
vrouter running ike-policy# aggressive true|false
```

## udp-encap

If true, enforce UDP encapsulation of ESP packets.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy  
vrouter running ike-policy# udp-encap true|false
```

## mobike

If true, enable MOBIKE (IKEv2 Mobility and Multihoming Protocol).

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy  
vrouter running ike-policy# mobike true|false
```

## ike-proposal

List of IKE phase 1 proposals.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy ike-proposal <uint8>
```

|         |   |
|---------|---|
| <uint8> | Index in the list of IKE phase 1 proposals. |
|---------|---|

## enc-alg

List of encryption algorithms for IKE SAs.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy ike-proposal <uint8>
vrouter running ike-proposal <uint8># enc-alg ENC-ALG
```

| ENC-ALG values  | Description                |
|-----------------|----------------------------|
| aes128-cbc      | AES-CBC, 128 bit key.      |
| aes192-cbc      | AES-CBC, 192 bit key.      |
| aes256-cbc      | AES-CBC, 256 bit key.      |
| des-cbc         | DES-CBC, 56 bit key.       |
| 3des-cbc        | 3DES-CBC, 168 bit key.     |
| aes128-ctr      | AES-CTR, 128 bit key.      |
| aes192-ctr      | AES-CTR, 192 bit key.      |
| aes256-ctr      | AES-CTR, 256 bit key.      |
| cast-cbc        | CAST-CBC, 128 bit key.     |
| blowfish128-cbc | Blowfish-CBC, 128 bit key. |
| blowfish192-cbc | Blowfish-CBC, 192 bit key. |
| blowfish256-cbc | Blowfish-CBC, 256 bit key. |
| camellia128-cbc | Camellia-CBC, 128 bit key. |
| camellia192-cbc | Camellia-CBC, 192 bit key. |
| camellia256-cbc | Camellia-CBC, 256 bit key. |
| camellia128-ctr | Camellia-CTR, 128 bit key. |
| camellia192-ctr | Camellia-CTR, 192 bit key. |
| camellia256-ctr | Camellia-CTR, 256 bit key. |

## auth-alg

List of auth algorithms for IKE SAs.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy ike-proposal <uint8>
vrouter running ike-proposal <uint8># auth-alg AUTH-ALG
```

| AUTH-ALG values | Description      |
|-----------------|------------------|
| hmac-md5        | HMAC-MD5-96.     |
| hmac-sha1       | HMAC-SHA1-96.    |
| hmac-sha256     | HMAC-SHA256-128. |
| hmac-sha384     | HMAC-SHA384-192. |
| hmac-sha512     | HMAC-SHA512-256. |
| aes-xcbc        | AES-XCBC-96.     |

## aead-alg

List of combined-mode (AEAD) algorithms for IKE SAs.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy ike-proposal <uint8>
vrouter running ike-proposal <uint8># aead-alg AEAD-ALG
```

| AEAD-ALG values    | Description                            |
|--------------------|--|
| aes128-gcm-64      | AES-GCM, 128 bit key, 64 bit ICV.      |
| aes192-gcm-64      | AES-GCM, 192 bit key, 64 bit ICV.      |
| aes256-gcm-64      | AES-GCM, 256 bit key, 64 bit ICV.      |
| aes128-gcm-96      | AES-GCM, 128 bit key, 96 bit ICV.      |
| aes192-gcm-96      | AES-GCM, 192 bit key, 96 bit ICV.      |
| aes256-gcm-96      | AES-GCM, 256 bit key, 96 bit ICV.      |
| aes128-gcm-128     | AES-GCM, 128 bit key, 128 bit ICV.     |
| aes192-gcm-128     | AES-GCM, 192 bit key, 128 bit ICV.     |
| aes256-gcm-128     | AES-GCM, 256 bit key, 128 bit ICV.     |
| aes128-ccm-64      | AES-CCM, 128 bit key, 64 bit ICV.      |
| aes192-ccm-64      | AES-CCM, 192 bit key, 64 bit ICV.      |
| aes256-ccm-64      | AES-CCM, 256 bit key, 64 bit ICV.      |
| aes128-ccm-96      | AES-CCM, 128 bit key, 96 bit ICV.      |
| aes192-ccm-96      | AES-CCM, 192 bit key, 96 bit ICV.      |
| aes256-ccm-96      | AES-CCM, 256 bit key, 96 bit ICV.      |
| aes128-ccm-128     | AES-CCM, 128 bit key, 128 bit ICV.     |
| aes192-ccm-128     | AES-CCM, 192 bit key, 128 bit ICV.     |
| aes256-ccm-128     | AES-CCM, 256 bit key, 128 bit ICV.     |
| camellia128-ccm-64 | Camellia-CCM, 128 bit key, 64 bit ICV. |
| camellia192-ccm-64 | Camellia-CCM, 192 bit key, 64 bit ICV. |
| camellia256-ccm-64 | Camellia-CCM, 256 bit key, 64 bit ICV. |
| camellia128-ccm-96 | Camellia-CCM, 128 bit key, 96 bit ICV. |
| camellia192-ccm-96 | Camellia-CCM, 192 bit key, 96 bit ICV. |
| camellia256-ccm-96 | Camellia-CCM, 256 bit key, 96 bit ICV. |

## prf-alg

List of pseudo-random algorithms for IKE SAs.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy ike-proposal <uint8>
vrouter running ike-proposal <uint8># prf-alg PRF-ALG
```

| PRF-ALG values | Description       |
|----------------|-------------------|
| hmac-md5       | PRF-HMAC-MD5.     |
| hmac-sha1      | PRF-HMAC-SHA1.    |
| aes-xcbc       | AES-XCBC-PRF-128. |
| aes-cmac       | AES-CMAC-PRF-128. |
| hmac-sha256    | PRF-HMAC-SHA-256. |
| hmac-sha384    | PRF-HMAC-SHA-384. |
| hmac-sha512    | PRF-HMAC-SHA-512. |

## dh-group

List of Diffie Hellman groups for key exchange.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ike-policy ike-proposal <uint8>
vrouter running ike-proposal <uint8># dh-group DH-GROUP
```

| DH-GROUP values | Description   |
|-----------------|---|
| modp768         | Modulo Prime 768 bits (group 1).                      |
| modp1024        | Modulo Prime 1024 bits (group 2).                     |
| modp1536        | Modulo Prime 1536 bits (group 5).                     |
| modp2048        | Modulo Prime 2048 bits (group 14).                    |
| modp3072        | Modulo Prime 3072 bits (group 15).                    |
| modp4096        | Modulo Prime 4096 bits (group 16).                    |
| modp6144        | Modulo Prime 6144 bits (group 17).                    |
| modp8192        | Modulo Prime 8192 bits (group 18).                    |
| modp1024s160    | Modulo Prime 1024 bits, Subgroup 160 bits (group 22). |
| modp1024s224    | Modulo Prime 1024 bits, Subgroup 224 bits (group 23). |
| modp1024s256    | Modulo Prime 1024 bits, Subgroup 256 bits (group 24). |
| ecp192          | Elliptic Curve 192 bits (group 25).                   |
| ecp224          | Elliptic Curve 224 bits (group 26).                   |
| ecp256          | Elliptic Curve 256 bits (group 19).                   |
| ecp384          | Elliptic Curve 384 bits (group 20).                   |
| ecp521          | Elliptic Curve 521 bits (group 21).                   |
| ecp224bp        | Brainpool Elliptic Curve 224 bits (group 27).         |
| ecp256bp        | Brainpool Elliptic Curve 256 bits (group 28).         |
| ecp384bp        | Brainpool Elliptic Curve 384 bits (group 29).         |
| ecp512bp        | Brainpool Elliptic Curve 512 bits (group 30).         |

## ipsec-policy

IPsec policy configuration.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy
```

### template (config only) (mandatory)

Template from which this IPsec policy derives.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy
vrouter running ipsec-policy# template <leafref>
```

### start-action

Action to perform for this CHILD\_SA on DPD timeout.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy
vrouter running ipsec-policy# start-action START-ACTION
```

| START-ACTION values | Description   |
|---------------------|---|
| none                | Load the connection only, can be used as a responder configuration.                             |
| trap                | Install a trap policy, which triggers the tunnel as soon as matching traffic has been detected. |
| start               | Initiate the connection actively.   |

### close-action

Action to perform when a CHILD\_SA gets closed by a peer.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy
vrouter running ipsec-policy# close-action CLOSE-ACTION
```

| CLOSE-ACTION values | Description  |
|---------------------|--|
| none                | Close the Child SA and take no further action.                                       |
| trap                | Install a trap policy matching traffic and try to re-negotiate the tunnel on-demand. |
| start               | Try to immediately re-create the CHILD_SA.   |

## dpd-action

Action to perform for a CHILD\_SA on DPD timeout.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy
vrouter running ipsec-policy# dpd-action DPD-ACTION
```

| DPD-ACTION values | Description   |
|-------------------|---|
| clear             | Close the Child SA and take no further action.  |
| trap              | Install a trap policy, which will catch matching traffic and tries to re-negotiate the tunnel on-demand action. |
| restart           | Immediately try to re-negotiate the CHILD_SA under a fresh IKE_SA.  |

## replay-window

Replay window size. 0 disables IPsec replay protection.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy
vrouter running ipsec-policy# replay-window <uint16>
```

## rekey-time

Time before initiating CHILD\_SA rekeying.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy
vrouter running ipsec-policy# rekey-time REKEY-TIME
```

|            |   |
|------------|---|
| REKEY-TIME | IKE duration, with optional unit (s m h d). |
|------------|---|

## life-time

Maximum lifetime before CHILD\_SA gets closed (default rekey-time + 10%).

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy
vrouter running ipsec-policy# life-time LIFE-TIME
```

|           |   |
|-----------|---|
| LIFE-TIME | IKE duration, with optional unit (s m h d). |
|-----------|---|

## rand-time

Time range from which to choose a random value to subtract from rekey\_time (default life\_time - rekey\_time).

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy
vrouter running ipsec-policy# rand-time RAND-TIME
```

|           |   |
|-----------|---|
| RAND-TIME | IKE duration, with optional unit (s m h d). |
|-----------|---|

## rekey-bytes

Number of bytes processed before initiating CHILD\_SA rekeying.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy
vrouter running ipsec-policy# rekey-bytes <uint64>
```

## life-bytes

Maximum bytes processed before CHILD\_SA gets closed (default rekey- bytes + 10%).

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy
vrouter running ipsec-policy# life-bytes <uint64>
```

## rand-bytes

Byte range from which to choose a random value to subtract from rekey\_bytes (default life\_bytes - rekey\_bytes).

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy
vrouter running ipsec-policy# rand-bytes <uint64>
```

## rekey-packets

Number of packets processed before initiating CHILD\_SA rekeying.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy
vrouter running ipsec-policy# rekey-packets <uint64>
```

### life-packets

Maximum packets processed before CHILD\_SA gets closed (default rekey- bytes + 10%).

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy  
vrouter running ipsec-policy# life-packets <uint64>
```

### rand-packets

Packet range from which to choose a random value to subtract from rekey\_packets (default life\_bytes - rekey\_bytes).

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy  
vrouter running ipsec-policy# rand-packets <uint64>
```

### encap-copy-dscp

Whether to copy DSCP from inner to outer IP header at IPsec encapsulation.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy  
vrouter running ipsec-policy# encap-copy-dscp true|false
```

### decap-copy-dscp

Whether to copy DSCP from outer to inner IP header at IPsec decapsulation.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy  
vrouter running ipsec-policy# decap-copy-dscp true|false
```

### encap-copy-df

Whether to copy the Don't Fragment bit from outer to inner IP header at IPsec encapsulation.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy  
vrouter running ipsec-policy# encap-copy-df true|false
```

## esp-proposal

List of ESP proposals.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy esp-proposal <uint8>
```

|         |                                 |
|---------|---------------------------------|
| <uint8> | Index in list of ESP proposals. |
|---------|---------------------------------|

## enc-alg

List of encryption algorithms for IPsec SAs.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy esp-proposal <uint8>
vrouter running esp-proposal <uint8># enc-alg ENC-ALG
```

| ENC-ALG values | Description            |
|----------------|------------------------|
| null           | NULL.                  |
| aes128-cbc     | AES-CBC, 128 bit key.  |
| aes192-cbc     | AES-CBC, 192 bit key.  |
| aes256-cbc     | AES-CBC, 256 bit key.  |
| des-cbc        | DES-CBC, 56 bit key.   |
| 3des-cbc       | 3DES-CBC, 168 bit key. |

## auth-alg

List of auth algorithms for IPsec SAs.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy esp-proposal <uint8>
vrouter running esp-proposal <uint8># auth-alg AUTH-ALG
```

| AUTH-ALG values | Description      |
|-----------------|------------------|
| none            | NONE.            |
| hmac-md5        | HMAC-MD5-96.     |
| hmac-sha1       | HMAC-SHA1-96.    |
| hmac-sha256     | HMAC-SHA256-128. |
| hmac-sha384     | HMAC-SHA384-192. |
| hmac-sha512     | HMAC-SHA512-256. |
| aes-xcbc        | AES-XCBC-96.     |

## aead-alg

List of combined-mode (AEAD) algorithms for IPsec SAs.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy esp-proposal <uint8>
vrouter running esp-proposal <uint8># aead-alg AEAD-ALG
```

| AEAD-ALG values | Description                         |
|-----------------|-------------------------------------|
| aes128-gcm-128  | AES-GCM, 128 bit key, 128 bit ICV.  |
| aes192-gcm-128  | AES-GCM, 192 bit key, 128 bit ICV.  |
| aes256-gcm-128  | AES-GCM, 256 bit key, 128 bit ICV.  |
| aes128-gmac     | AES-GMAC, 128 bit key, 128 bit ICV. |
| aes192-gmac     | AES-GMAC, 192 bit key, 128 bit ICV. |
| aes256-gmac     | AES-GMAC, 256 bit key, 128 bit ICV. |

## dh-group

List of Diffie Hellman groups for Perfect Forward Secrecy.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy esp-proposal <uint8>
vrouter running esp-proposal <uint8># dh-group DH-GROUP
```

| DH-GROUP values | Description   |
|-----------------|---|
| modp768         | Modulo Prime 768 bits (group 1).                      |
| modp1024        | Modulo Prime 1024 bits (group 2).                     |
| modp1536        | Modulo Prime 1536 bits (group 5).                     |
| modp2048        | Modulo Prime 2048 bits (group 14).                    |
| modp3072        | Modulo Prime 3072 bits (group 15).                    |
| modp4096        | Modulo Prime 4096 bits (group 16).                    |
| modp6144        | Modulo Prime 6144 bits (group 17).                    |
| modp8192        | Modulo Prime 8192 bits (group 18).                    |
| modp1024s160    | Modulo Prime 1024 bits, Subgroup 160 bits (group 22). |
| modp1024s224    | Modulo Prime 1024 bits, Subgroup 224 bits (group 23). |
| modp1024s256    | Modulo Prime 1024 bits, Subgroup 256 bits (group 24). |
| ecp192          | Elliptic Curve 192 bits (group 25).                   |
| ecp224          | Elliptic Curve 224 bits (group 26).                   |
| ecp256          | Elliptic Curve 256 bits (group 19).                   |
| ecp384          | Elliptic Curve 384 bits (group 20).                   |
| ecp521          | Elliptic Curve 521 bits (group 21).                   |
| ecp224bp        | Brainpool Elliptic Curve 224 bits (group 27).         |
| ecp256bp        | Brainpool Elliptic Curve 256 bits (group 28).         |
| ecp384bp        | Brainpool Elliptic Curve 384 bits (group 29).         |
| ecp512bp        | Brainpool Elliptic Curve 512 bits (group 30).         |

## esn

List of Extended Sequence Number modes.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy esp-proposal <uint8>
vrouter running esp-proposal <uint8># esn true|false
```

## ah-proposal

List of AH proposals.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy ah-proposal <uint8>
```

|         |                                |
|---------|--------------------------------|
| <uint8> | Index in list of AH proposals. |
|---------|--------------------------------|

## auth-alg

List of auth algorithms for IPsec SAs.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy ah-proposal <uint8>
vrouter running ah-proposal <uint8># auth-alg AUTH-ALG
```

| AUTH-ALG values | Description      |
|-----------------|------------------|
| hmac-md5        | HMAC-MD5-96.     |
| hmac-sha1       | HMAC-SHA1-96.    |
| hmac-sha256     | HMAC-SHA256-128. |
| hmac-sha384     | HMAC-SHA384-192. |
| hmac-sha512     | HMAC-SHA512-256. |
| aes-xcbc        | AES-XCBC-96.     |

## dh-group

List of Diffie Hellman groups for Perfect Forward Secrecy.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy ah-proposal <uint8>
vrouter running ah-proposal <uint8># dh-group DH-GROUP
```

| DH-GROUP values | Description   |
|-----------------|---|
| modp768         | Modulo Prime 768 bits (group 1).                      |
| modp1024        | Modulo Prime 1024 bits (group 2).                     |
| modp1536        | Modulo Prime 1536 bits (group 5).                     |
| modp2048        | Modulo Prime 2048 bits (group 14).                    |
| modp3072        | Modulo Prime 3072 bits (group 15).                    |
| modp4096        | Modulo Prime 4096 bits (group 16).                    |
| modp6144        | Modulo Prime 6144 bits (group 17).                    |
| modp8192        | Modulo Prime 8192 bits (group 18).                    |
| modp1024s160    | Modulo Prime 1024 bits, Subgroup 160 bits (group 22). |
| modp1024s224    | Modulo Prime 1024 bits, Subgroup 224 bits (group 23). |
| modp1024s256    | Modulo Prime 1024 bits, Subgroup 256 bits (group 24). |
| ecp192          | Elliptic Curve 192 bits (group 25).                   |
| ecp224          | Elliptic Curve 224 bits (group 26).                   |
| ecp256          | Elliptic Curve 256 bits (group 19).                   |
| ecp384          | Elliptic Curve 384 bits (group 20).                   |
| ecp521          | Elliptic Curve 521 bits (group 21).                   |
| ecp224bp        | Brainpool Elliptic Curve 224 bits (group 27).         |
| ecp256bp        | Brainpool Elliptic Curve 256 bits (group 28).         |
| ecp384bp        | Brainpool Elliptic Curve 384 bits (group 29).         |
| ecp512bp        | Brainpool Elliptic Curve 512 bits (group 30).         |

## esn

List of Extended Sequence Number modes.

```
vrouter running config# vrf <vrf> ike vpn <vpn> ipsec-policy ah-proposal <uint8>
vrouter running ah-proposal <uint8># esn true|false
```

## security-policy

List of IPsec bidirectional security policies.

```
vrouter running config# vrf <vrf> ike vpn <vpn> security-policy <security-policy>
```

|                   |                       |
|-------------------|-----------------------|
| <security-policy> | IKE object name type. |
|-------------------|-----------------------|

## svti-id-in

SVTI ID set on inbound policies/SA.

```
vrouter running config# vrf <vrf> ike vpn <vpn> security-policy <security-policy>
vrouter running security-policy <security-policy># svti-id-in <uint32>
```

## svti-id-out

SVTI ID set on outbound policies/SA.

```
vrouter running config# vrf <vrf> ike vpn <vpn> security-policy <security-policy>
vrouter running security-policy <security-policy># svti-id-out <uint32>
```

## action

IPsec action.

```
vrouter running config# vrf <vrf> ike vpn <vpn> security-policy <security-policy>
vrouter running security-policy <security-policy># action ACTION
```

| ACTION values | Description  |
|---------------|--|
| esp           | Protect traffic with Encapsulating Security Payload. |
| ah            | Protect traffic with Authentication Header.          |
| pass          | Pass traffic in plain text.                          |
| drop          | Drop traffic.  |

## Default value

esp

## mode

IPsec mode if action is esp or ah.

```
vrouter running config# vrf <vrf> ike vpn <vpn> security-policy <security-policy>
vrouter running security-policy <security-policy># mode MODE
```

| MODE values | Description                   |
|-------------|-------------------------------|
| tunnel      | Tunnel mode.                  |
| transport   | Transport mode.               |
| beet        | Bound End to End Tunnel mode. |

**Default value**

tunnel

**priority**

Security policy priority (0 stands for dynamically calculated).

```
vrouter running config# vrf <vrf> ike vpn <vpn> security-policy <security-policy>
vrouter running security-policy <security-policy># priority <uint32>
```

**Default value**

0

**local-ts**

Local traffic selector (default the tunnel outer address or the virtual IP, if negotiated).

```
vrouter running config# vrf <vrf> ike vpn <vpn> security-policy <security-policy>
vrouter running security-policy <security-policy># local-ts subnet SUBNET \
... protocol <uint8> port <uint16>
```

**subnet**

Private subnet or address (default: the tunnel outer address or virtual IP, if negotiated).

```
subnet SUBNET
```

| SUBNET values  | Description                            |
|----------------|--|
| <ipv4-address> | An IPv4 address.                       |
| <ipv6-address> | An IPv6 address.                       |
| <ipv4-prefix>  | An IPv4 prefix: address and CIDR mask. |
| <ipv6-prefix>  | An IPv6 prefix: address and CIDR mask. |

**protocol**

Protocol number (default any).

```
protocol <uint8>
```

**port**

Port number or ICMP type/code (default any).

```
port <uint16>
```

**remote-ts**

Remote traffic selector (default the tunnel outer address or the virtual IP, if negotiated).

```
vrouter running config# vrf <vrf> ike vpn <vpn> security-policy <security-policy>
vrouter running security-policy <security-policy># remote-ts subnet SUBNET \
... protocol <uint8> port <uint16>
```

**subnet**

Private subnet or address (default: the tunnel outer address or virtual IP, if negotiated).

```
subnet SUBNET
```

| SUBNET values  | Description                            |
|----------------|--|
| <ipv4-address> | An IPv4 address.                       |
| <ipv6-address> | An IPv6 address.                       |
| <ipv4-prefix>  | An IPv4 prefix: address and CIDR mask. |
| <ipv6-prefix>  | An IPv6 prefix: address and CIDR mask. |

**protocol**

Protocol number (default any).

```
protocol <uint8>
```

**port**

Port number or ICMP type/code (default any).

```
port <uint16>
```

**ike-sas (state only)**

Number of IKE SAs.

**total (state only)**

Total number of IKE SAs (half-open or established).

```
vrouter> show state vrf <vrf> ike ike-sas total
```

**half-open (state only)**

Number of half-open IKE SAs.

```
vrouter> show state vrf <vrf> ike ike-sas half-open
```

**task-processing (state only)**

Internal task processing statistics.

**worker-threads (state only)**

State of IKE daemon threads.

**total (state only)**

Total number of threads.

```
vrouter> show state vrf <vrf> ike task-processing worker-threads total
```

**idle (state only)**

Number of idle threads.

```
vrouter> show state vrf <vrf> ike task-processing worker-threads idle
```

**critical (state only)**

Number of threads executing critical priority tasks.

```
vrouter> show state vrf <vrf> ike task-processing worker-threads critical
```

**high (state only)**

Number of threads executing high priority tasks.

```
vrouter> show state vrf <vrf> ike task-processing worker-threads high
```

**medium (state only)**

Number of threads executing medium priority tasks.

```
vrouter> show state vrf <vrf> ike task-processing worker-threads medium
```

**low (state only)**

Number of threads executing low priority tasks.

```
vrouter> show state vrf <vrf> ike task-processing worker-threads low
```

**task-queues (state only)**

Counters of pending tasks.

**critical (state only)**

Number of critical priority tasks waiting for an available thread.

```
vrouter> show state vrf <vrf> ike task-processing task-queues critical
```

**high (state only)**

Number of high priority tasks waiting for an available thread.

```
vrouter> show state vrf <vrf> ike task-processing task-queues high
```

**medium (state only)**

Number of medium priority tasks waiting for an available thread.

```
vrouter> show state vrf <vrf> ike task-processing task-queues medium
```

**low (state only)**

Number of low priority tasks waiting for an available thread.

```
vrouter> show state vrf <vrf> ike task-processing task-queues low
```

**scheduled (state only)**

Number of tasks waiting for a timer to expire.

```
vrouter> show state vrf <vrf> ike task-processing task-queues scheduled
```

**counters (state only)**

Global IKE message counters.

### ike-rekey-init (state only)

Initiated IKE\_SA rekeyings.

```
vrouter> show state vrf <vrf> ike counters ike-rekey-init
```

### ike-rekey-resp (state only)

Responded IKE\_SA rekeyings.

```
vrouter> show state vrf <vrf> ike counters ike-rekey-resp
```

### child-rekey (state only)

Completed CHILD\_SA rekeyings.

```
vrouter> show state vrf <vrf> ike counters child-rekey
```

### invalid (state only)

Messages with an invalid IKE SPI.

```
vrouter> show state vrf <vrf> ike counters invalid
```

### invalid-spi (state only)

Messages with invalid types, length, or a value out of range.

```
vrouter> show state vrf <vrf> ike counters invalid-spi
```

### ike-init-in-req (state only)

Received IKE\_SA\_INIT requests.

```
vrouter> show state vrf <vrf> ike counters ike-init-in-req
```

### ike-init-in-resp (state only)

Received IKE\_SA\_INIT responses.

```
vrouter> show state vrf <vrf> ike counters ike-init-in-resp
```

### ike-init-out-req (state only)

Sent IKE\_SA\_INIT requests.

```
vrouter> show state vrf <vrf> ike counters ike-init-out-req
```

### ike-init-out-resp (state only)

Sent IKE\_SA\_INIT responses.

```
vrouter> show state vrf <vrf> ike counters ike-init-out-resp
```

### ike-auth-in-req (state only)

Received IKE\_AUTH requests.

```
vrouter> show state vrf <vrf> ike counters ike-auth-in-req
```

### ike-auth-in-resp (state only)

Received IKE\_AUTH responses.

```
vrouter> show state vrf <vrf> ike counters ike-auth-in-resp
```

### ike-auth-out-req (state only)

Sent IKE\_AUTH requests.

```
vrouter> show state vrf <vrf> ike counters ike-auth-out-req
```

**ike-auth-out-req (state only)**

Sent IKE\_AUTH responses.

```
vrouter> show state vrf <vrf> ike counters ike-auth-out-req
```

**create-child-in-req (state only)**

Received CREATE\_CHILD\_SA requests.

```
vrouter> show state vrf <vrf> ike counters create-child-in-req
```

**create-child-in-req (state only)**

Received CREATE\_CHILD\_SA responses.

```
vrouter> show state vrf <vrf> ike counters create-child-in-req
```

**create-child-out-req (state only)**

Sent CREATE\_CHILD\_SA requests.

```
vrouter> show state vrf <vrf> ike counters create-child-out-req
```

**create-child-out-req (state only)**

Sent CREATE\_CHILD\_SA responses.

```
vrouter> show state vrf <vrf> ike counters create-child-out-req
```

**info-in-req (state only)**

Received INFORMATIONAL requests.

```
vrouter> show state vrf <vrf> ike counters info-in-req
```

### info-in-resp (state only)

Received INFORMATIONAL responses.

```
vrouter> show state vrf <vrf> ike counters info-in-resp
```

### info-out-req (state only)

Sent INFORMATIONAL requests.

```
vrouter> show state vrf <vrf> ike counters info-out-req
```

### info-out-resp (state only)

Sent INFORMATIONAL responses.

```
vrouter> show state vrf <vrf> ike counters info-out-resp
```

### vpn-counters (state only)

List of per-VPN IKE message counters.

### ike-rekey-init (state only)

Initiated IKE\_SA rekeyings.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> ike-rekey-init
```

### ike-rekey-resp (state only)

Responded IKE\_SA rekeyings.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> ike-rekey-resp
```

### child-rekey (state only)

Completed CHILD\_SA rekeyings.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> child-rekey
```

### invalid (state only)

Messages with an invalid IKE SPI.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> invalid
```

### invalid-spi (state only)

Messages with invalid types, length, or a value out of range.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> invalid-spi
```

### ike-init-in-req (state only)

Received IKE\_SA\_INIT requests.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> ike-init-in-req
```

### ike-init-in-resp (state only)

Received IKE\_SA\_INIT responses.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> ike-init-in-resp
```

### ike-init-out-req (state only)

Sent IKE\_SA\_INIT requests.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> ike-init-out-req
```

### ike-init-out-resp (state only)

Sent IKE\_SA\_INIT responses.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> ike-init-out-resp
```

### ike-auth-in-req (state only)

Received IKE\_AUTH requests.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> ike-auth-in-req
```

### ike-auth-in-resp (state only)

Received IKE\_AUTH responses.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> ike-auth-in-resp
```

### ike-auth-out-req (state only)

Sent IKE\_AUTH requests.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> ike-auth-out-req
```

### ike-auth-out-resp (state only)

Sent IKE\_AUTH responses.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> ike-auth-out-resp
```

### create-child-in-req (state only)

Received CREATE\_CHILD\_SA requests.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> create-child-in-req
```

**create-child-in-resp (state only)**

Received CREATE\_CHILD\_SA responses.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> create-child-in-resp
```

**create-child-out-req (state only)**

Sent CREATE\_CHILD\_SA requests.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> create-child-out-req
```

**create-child-out-resp (state only)**

Sent CREATE\_CHILD\_SA responses.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> create-child-out-  
↳resp
```

**info-in-req (state only)**

Received INFORMATIONAL requests.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> info-in-req
```

**info-in-resp (state only)**

Received INFORMATIONAL responses.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> info-in-resp
```

**info-out-req (state only)**

Sent INFORMATIONAL requests.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> info-out-req
```

### info-out-resp (state only)

Sent INFORMATIONAL responses.

```
vrouter> show state vrf <vrf> ike vpn-counters name <vpn-counters> info-out-resp
```

### ike-sa (state only)

List of IKE Security Associations.

#### name (state only)

Name of the VPN.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> name
```

#### version (state only)

IKE version.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> version
```

#### state (state only)

IKE SA state.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> state
```

#### local-address (state only)

Local IKE IP address.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> local-address
```

**remote-address (state only)**

Remote IKE IP address.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> remote-address
```

**local-port (state only)**

Local IKE UDP port.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> local-port
```

**remote-port (state only)**

Remote IKE UDP port.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> remote-port
```

**local-id (state only)**

Local IKE identifier.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> local-id
```

**remote-id (state only)**

Remote IKE identifier.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> remote-id
```

**remote-eap-id (state only)**

Remote EAP identifier.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> remote-eap-id
```

### initiator-spi (state only)

IKE initiator SPI.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> initiator-spi
```

### responder-spi (state only)

IKE responder SPI.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> responder-spi
```

### enc-alg (state only)

IKE encryption algorithm.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> enc-alg
```

### auth-alg (state only)

IKE authentication algorithm.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> auth-alg
```

### aead-alg (state only)

IKE combined-mode algorithm.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> aead-alg
```

### prf-alg (state only)

IKE pseudo-random algorithm.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> prf-alg
```

### dh-group (state only)

IKE Diffie Hellman group.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> dh-group
```

### established-time (state only)

Seconds since IKE session was established.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> established-time
```

### rekey-time (state only)

Seconds before IKE session is rekeyed.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> rekey-time
```

### reauth-time (state only)

Seconds before IKE session is reauthenticated.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> reauth-time
```

### udp-encap (state only)

UDP encapsulation state.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> udp-encap
```

### mobike (state only)

IKEv2 Mobility and Multihoming Protocol (MOBIKE) state.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> mobike
```

### local-vip (state only)

List of local virtual IP addresses.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> local-vip
```

### remote-vip (state only)

List of local virtual IP addresses.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> remote-vip
```

### child-sa (state only)

List of Child Security Associations.

#### name (state only)

Name of the policy.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ name
```

#### state (state only)

Child SA state.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ state
```

#### reqid (state only)

Request ID of the Child SA, that binds IPsec SAs to SPs.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ reqid
```

### protocol (state only)

IPsec protocol.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ protocol
```

### udp-encap (state only)

UDP encapsulation state.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ udp-encap
```

### mobike (state only)

IPv2 Mobility and Multihoming Protocol (MOBIKE) state.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ mobike
```

### spi-in (state only)

Inbound Security Parameters Index.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ spi-in
```

### spi-out (state only)

Outbound Security Parameters Index.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ spi-out
```

### svti-id-in (state only)

SVTI ID set on inbound SA.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ svti-id-in
```

### svti-id-out (state only)

SVTI ID set on outbound SA.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ svti-id-out
```

### enc-alg (state only)

ESP encryption algorithm.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ enc-alg
```

### auth-alg (state only)

ESP or AH authentication algorithm.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ auth-alg
```

### aead-alg (state only)

ESP combined-mode algorithm.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ aead-alg
```

### dh-group (state only)

Diffie Hellman group for Perfect Forward Secrecy.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↳ dh-group
```

### esn (state only)

Extended Sequence Number state.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↳ esn
```

### bytes-in (state only)

Input bytes processed by this Child SA.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↳ bytes-in
```

### packets-in (state only)

Input packets processed by this Child SA.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↳ packets-in
```

### bytes-out (state only)

Output bytes processed by this Child SA.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↳ bytes-out
```

### packets-out (state only)

Output packets processed by this Child SA.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ packets-out
```

### installed-time (state only)

Seconds since IPsec SAs were installed.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ installed-time
```

### rekey-time (state only)

Seconds before IPsec SAs are rekeyed.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ rekey-time
```

### life-time (state only)

Seconds before IPsec SAs are deleted.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ life-time
```

### mode (state only)

IPsec mode.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ mode
```

### local-ts (state only)

Local traffic selector.

### subnet (state only)

Private subnet or address (default: the tunnel outer address or virtual IP, if negotiated).

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ local-ts subnet
```

### protocol (state only)

Protocol number (default any).

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ local-ts protocol
```

### port (state only)

Port number or ICMP type/code (default any).

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ local-ts port
```

### unsupported (state only)

The type of traffic selector proposed by the remote peer is not supported. The configuration may not work as expected.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ local-ts unsupported
```

**remote-ts (state only)**

Remote traffic selector.

**subnet (state only)**

Private subnet or address (default: the tunnel outer address or virtual IP, if negotiated).

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ remote-ts subnet
```

**protocol (state only)**

Protocol number (default any).

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ remote-ts protocol
```

**port (state only)**

Port number or ICMP type/code (default any).

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ remote-ts port
```

**unsupported (state only)**

The type of traffic selector proposed by the remote peer is not supported. The configuration may not work as expected.

```
vrouter> show state vrf <vrf> ike ike-sa unique-id <uint32> child-sa unique-id <uint32>  
↪ remote-ts unsupported
```

### pool-lease (state only)

List of virtual address pool leases.

### address (state only)

First virtual address in the pool.

```
vrouter> show state vrf <vrf> ike pool-lease name <pool-lease> address
```

### size (state only)

Virtual address pool size.

```
vrouter> show state vrf <vrf> ike pool-lease name <pool-lease> size
```

### online (state only)

Number of online virtual addresses.

```
vrouter> show state vrf <vrf> ike pool-lease name <pool-lease> online
```

### offline (state only)

Number of offline virtual addresses.

```
vrouter> show state vrf <vrf> ike pool-lease name <pool-lease> offline
```

## 3.2.26 sflow

---

**Note:** requires a Turbo Router Network License.

---

SFlow configuration.

```
vrouter running config# vrf <vrf> sflow
```

## enabled

Enable or disable the sFlow daemon for perf measurement.

```
vrouter running config# vrf <vrf> sflow
vrouter running sflow# enabled true|false
```

### Default value

true

## agent-interface

Use this interface IP in the reports sent to the collector.

```
vrouter running config# vrf <vrf> sflow
vrouter running sflow# agent-interface <string>
```

## polling

Polling type (disabled or interval in seconds). Every interval, an sFlow frame containing interface statistics is sent to the collector.

```
vrouter running config# vrf <vrf> sflow
vrouter running sflow# polling POLLING
```

| POLLING values | Description                  |
|----------------|------------------------------|
| disabled       | Polling disabled.            |
| <uint32>       | Polling interval in seconds. |

### Default value

disabled

## sflow-port

The port number to receive sFlow sample from 6WIND products.

```
vrouter running config# vrf <vrf> sflow
vrouter running sflow# sflow-port SFLOW-PORT
```

|            |   |
|------------|---|
| SFLOW-PORT | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|------------|---|

### Default value

36343

**if-error**

Force the output ifindex value used for drop packets.

```
vrouter running config# vrf <vrf> sflow
vrouter running sflow# if-error IF-ERROR
```

| IF-ERROR values | Description               |
|-----------------|---------------------------|
| 1073741823      | Last index 0x3FFFFFFF.    |
| 1073741824      | Generic error 0x40000000. |
| <uint32>        | No description.           |

**if-unknown**

Force the output ifindex value used for packets where the output is not known.

```
vrouter running config# vrf <vrf> sflow
vrouter running sflow# if-unknown IF-UNKNOWN
```

| IF-UNKNOWN values | Description            |
|-------------------|------------------------|
| 1073741823        | Last index 0x3FFFFFFF. |
| 0                 | Generic error 0.       |
| <uint32>          | No description.        |

**sflow-collector**

List of sFlow collectors.

```
vrouter running config# vrf <vrf> sflow
vrouter running sflow# sflow-collector <sflow-collector> port PORT
```

|                |   |
|----------------|---|
| <sflow-values> | Description   |
| <A.B.C.D>      | IPv4 address.   |
| <X:X::X:X>     | IPv6 address.   |
| <host-name>    | <p>The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492.</p> |

### port

The port number of the sFlow collector.

|           |
|-----------|
| port PORT |
|-----------|

|      |   |
|------|---|
| PORT | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|------|---|

### Default value

6343

### sflow-interface

List of sFlow interfaces.

|   |
|---|
| <b>vrouter running config#</b> vrf <vrf> sflow                  |
| <b>vrouter running sflow#</b> sflow-interface <sflow-interface> |

|                   |                    |
|-------------------|--------------------|
| <sflow-interface> | An interface name. |
|-------------------|--------------------|

## sflow-sampling

List of sampling rate by interface speed.

```
vrouter running config# vrf <vrf> sflow
vrouter running sflow# sflow-sampling speed <sflow-sampling> rate RATE
```

| <sflow-sampling> values | Description                                       |
|-------------------------|---|
| <string>                | Custom speed interfaces with the xxx[M G] format. |
| 100M                    | 100Mbps interface.                                |
| 1G                      | 1Gbps interface.                                  |
| 10G                     | 10Gbps interface.                                 |
| 40G                     | 40Gbps interface.                                 |
| 100G                    | 100Gbps interface.                                |
| other                   | Interface with no speed.                          |

## rate

Sampling rate in number of packets. For better performance, it should be set to a power of two.

```
rate RATE
```

| RATE values | Description                            |
|-------------|--|
| auto        | Automatically derived from link speed. |
| <uint32>    | SFlow sampling rate.                   |

### Default value

auto

## 3.2.27 snmp

---

**Note:** requires a Turbo Router Network License.

---

SNMP configuration.

```
vrouter running config# vrf <vrf> snmp
```

## enabled

Enable or disable the SNMP engine.

```
vrouter running config# vrf <vrf> snmp
vrouter running snmp# enabled true|false
```

## Default value

true

## listen

Configuration of the transport endpoint on which the engine listens.

```
vrouter running config# vrf <vrf> snmp listen
```

## protocols

The protocols used for connecting to the SNMP agent.

```
vrouter running config# vrf <vrf> snmp listen
vrouter running listen# protocols PROTOCOLS
```

| PROTOCOLS values | Description |
|------------------|-------------|
| udp              | UDP.        |
| tcp              | TCP.        |
| udp6             | UDPv6.      |
| tcp6             | TCPv6.      |

## Default value

udp

## port

The TCP or UDP port on which the engine listens.

```
vrouter running config# vrf <vrf> snmp listen
vrouter running listen# port PORT
```

|      |   |
|------|---|
| PORT | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|------|---|

**Default value**

161

**static-info**

Most of the information reported by the SNMP agent is retrieved from the underlying system. However, certain MIB objects can be configured with a static value.

```
vrouter running config# vrf <vrf> snmp static-info
```

**location**

System location (sysLocation.0) object value.

```
vrouter running config# vrf <vrf> snmp static-info  
vrouter running static-info# location <string>
```

**contact**

System contact (sysContact.0) object value.

```
vrouter running config# vrf <vrf> snmp static-info  
vrouter running static-info# contact <string>
```

**name**

System name (sysName.0) object value.

```
vrouter running config# vrf <vrf> snmp static-info  
vrouter running static-info# name <string>
```

**services**

Value of the sysServices.0 object. For a host system, a good value is 72 (application + end-to-end layers).

```
vrouter running config# vrf <vrf> snmp static-info  
vrouter running static-info# services <uint8>
```

## description

System description of the SNMP agent (sysDescr.0).

```
vrouter running config# vrf <vrf> snmp static-info
vrouter running static-info# description <string>
```

## object-id

System OID (sysObjectOID.0) object value.

```
vrouter running config# vrf <vrf> snmp static-info
vrouter running static-info# object-id OBJECT-ID
```

|           |   |
|-----------|---|
| OBJECT-ID | SNMP object identifier either as a label or numeric form. |
|-----------|---|

## view

A named 'view' - a subset of the overall OID tree.

```
vrouter running config# vrf <vrf> snmp view <string>
```

|          |                       |
|----------|-----------------------|
| <string> | The name of the view. |
|----------|-----------------------|

## subtree

A part of the OID tree to include or exclude from the view.

```
vrouter running config# vrf <vrf> snmp view <string>
vrouter running view <string># subtree <subtree> included true|false
```

|           |   |
|-----------|---|
| <subtree> | SNMP object identifier either as a label or numeric form. |
|-----------|---|

**included**

Set to false to exclude this OID from the view.

```
included true|false
```

**Default value**

true

**community**

An SNMPv1 or SNMPv2c community.

```
vrouter running config# vrf <vrf> snmp community <string>
```

|          |                            |
|----------|----------------------------|
| <string> | The name of the community. |
|----------|----------------------------|

**authorization (mandatory)**

The authorization level of the community.

```
vrouter running config# vrf <vrf> snmp community <string>
vrouter running community <string># authorization AUTHORIZATION
```

| AUTHORIZATION values | Description                               |
|----------------------|---|
| read-only            | Read-only (GET and GETNEXT) access.       |
| read-write           | Read-write (GET, GETNEXT and SET) access. |

**source**

Restrict access to requests from the specified address or prefix list.

```
vrouter running config# vrf <vrf> snmp community <string>
vrouter running community <string># source SOURCE
```

| SOURCE values | Description  |
|---------------|--|
| <A.B.C.D>     | IPv4 address.  |
| <X:X::X:X>    | IPv6 address.  |
| <host-name>   | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D/M>   | IPv4 prefix: address and CIDR mask.  |
| <X:X::X:X/M>  | IPv6 prefix: address and CIDR mask.  |

### view

Restricts access for that community to the subtree rooted at the given view name. If not specified, the community has access to the whole OID tree.

```
vrouter running config# vrf <vrf> snmp community <string>
vrouter running community <string># view <leafref>
```

### monitored-vrf

Monitored VRF.

```
vrouter running config# vrf <vrf> snmp monitored-vrf <string>
```

|          |                                |
|----------|--------------------------------|
| <string> | The name of the monitored VRF. |
|----------|--------------------------------|

## identifier

Identifier to access the monitored VRF, acts as a community for SNMPv1 or SNMPv2c and as a context for SNMPv3.

```
vrouter running config# vrf <vrf> snmp monitored-vrf <string> identifier <string>
```

|          |  |
|----------|--|
| <string> | The monitored VRF identifier (community for SNMPv1 or SNMPv2c and context for SNMPv3). |
|----------|--|

## authorization (mandatory)

The authorization level of the identifier.

```
vrouter running config# vrf <vrf> snmp monitored-vrf <string> identifier <string>  
vrouter running identifier <string># authorization AUTHORIZATION
```

| AUTHORIZATION values | Description                               |
|----------------------|---|
| read-only            | Read-only (GET and GETNEXT) access.       |
| read-write           | Read-write (GET, GETNEXT and SET) access. |

## source

Restrict access to requests from the specified address or prefix list for SNMPv1 or SNMPv2.

```
vrouter running config# vrf <vrf> snmp monitored-vrf <string> identifier <string>  
vrouter running identifier <string># source SOURCE
```

| SOURCE values | Description  |
|---------------|--|
| <A.B.C.D>     | IPv4 address.  |
| <X:X::X:X>    | IPv6 address.  |
| <host-name>   | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |
| <A.B.C.D/M>   | prefix: address and CIDR mask.   |
| <X:X::X:X/M>  | prefix: address and CIDR mask.   |

### view

Restricts access to the subtree rooted at the given view name. If not specified, the identifier has access to the whole OID tree.

```
vrouter running config# vrf <vrf> snmp monitored-vrf <string> identifier <string>
vrouter running identifier <string># view <leafref>
```

### traps

Active monitoring and automatic notifications configuration.

```
vrouter running config# vrf <vrf> snmp monitored-vrf <string> traps
```

## destination

The destination of SNMPv1 TRAPs, SNMPv2c TRAP2s, or SNMPv2 INFORM notifications.

```
vrouter running config# vrf <vrf> snmp monitored-vrf <string> traps destination  
↳<leafref>
```

|           |                              |
|-----------|------------------------------|
| <leafref> | The receiver address to use. |
|-----------|------------------------------|

## community (mandatory)

The community string to use when sending traps to this destination.

```
vrouter running config# vrf <vrf> snmp monitored-vrf <string> traps destination  
↳<leafref>  
vrouter running destination <leafref># community <leafref>
```

## access-control

SNMPv3 access control configuration.

```
vrouter running config# vrf <vrf> snmp access-control
```

## user

An SNMPv3 user.

```
vrouter running config# vrf <vrf> snmp access-control user <string>
```

|          |                                      |
|----------|--------------------------------------|
| <string> | The name of the user (securityName). |
|----------|--------------------------------------|

## auth-password (mandatory)

The authentication password.

```
vrouter running config# vrf <vrf> snmp access-control user <string>  
vrouter running user <string># auth-password <string>
```

## auth-method

The authentication method.

```
vrouter running config# vrf <vrf> snmp access-control user <string>
vrouter running user <string># auth-method AUTH-METHOD
```

| AUTH-METHOD values | Description |
|--------------------|-------------|
| md5                | MD5.        |
| sha                | SHA.        |

### Default value

sha

## priv-password

The privacy (encryption) password. If not specified, it is assumed to be the same as the authentication password.

```
vrouter running config# vrf <vrf> snmp access-control user <string>
vrouter running user <string># priv-password <string>
```

## priv-protocol

The encryption protocol.

```
vrouter running config# vrf <vrf> snmp access-control user <string>
vrouter running user <string># priv-protocol PRIV-PROTOCOL
```

| PRIV-PROTOCOL values | Description |
|----------------------|-------------|
| aes                  | AES.        |
| des                  | DES.        |

### Default value

aes

## group

An SNMPv3 group.

```
vrouter running config# vrf <vrf> snmp access-control group <string>
```

|          |                        |
|----------|------------------------|
| <string> | The name of the group. |
|----------|------------------------|

## user

Name of a user to add to this group.

```
vrouter running config# vrf <vrf> snmp access-control group <string>
vrouter running group <string># user <leafref>
```

## security-level (mandatory)

The security level enforced on this group.

```
vrouter running config# vrf <vrf> snmp access-control group <string>
vrouter running group <string># security-level SECURITY-LEVEL
```

| SECURITY-LEVEL values | Description                                 |
|-----------------------|---|
| auth                  | Authentication is required.                 |
| priv                  | Authentication and encryption are required. |

## view

Restricts access for that group to the subtree rooted at the given view name. If not specified, the group has access to the whole OID tree.

```
vrouter running config# vrf <vrf> snmp access-control group <string>
vrouter running group <string># view <leafref>
```

## authorization

The authorization level of this group.

```
vrouter running config# vrf <vrf> snmp access-control group <string>
vrouter running group <string># authorization AUTHORIZATION
```

| AUTHORIZATION values | Description                               |
|----------------------|---|
| read-only            | Read-only (GET and GETNEXT) access.       |
| read-write           | Read-write (GET, GETNEXT and SET) access. |

### Default value

read-only

## traps

Active monitoring and automatic notifications configuration.

```
vrouter running config# vrf <vrf> snmp traps
```

## destination

Notification receiver that should be sent SNMPv1 TRAPs, SNMPv2c TRAP2s, or SNMPv2 INFORM notifications.

```
vrouter running config# vrf <vrf> snmp traps
vrouter running traps# destination <destination> port PORT protocol PROTOCOL \
... notification-type NOTIFICATION-TYPE community <leafref>
```

| <dest-values | Description  |
|--------------|--|
| <A.B.C.D>    | IPv4 address.  |
| <X:X::X:X>   | IPv6 address.  |
| <host-name>  | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |

## port

The port number of the host where to send the traps.

```
port PORT
```

|      |   |
|------|---|
| PORT | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|------|---|

## Default value

162

## protocol

The protocol used to connect to the destination host.

```
protocol PROTOCOL
```

| PROTOCOL values | Description |
|-----------------|-------------|
| udp             | UDP.        |
| tcp             | TCP.        |
| udp6            | UDPv6.      |
| tcp6            | TCPv6.      |

**Default value**

udp

**notification-type (mandatory)**

The type of notifications that is to be sent to the specified host.

```
notification-type NOTIFICATION-TYPE
```

| NOTIFICATION-TYPE values | Description   |
|--------------------------|---|
| TRAP                     | Send SNMPv1 TRAPs to the specified host.                |
| TRAP2                    | Send SNMPv2c TRAP2s to the specified host.              |
| INFORM                   | Send SNMPv2 INFORM notifications to the specified host. |

**community (mandatory)**

The community string to use when sending traps to this destination.

```
community <leafref>
```

**authfail-check**

Monitor authentication failures.

```
vrouter running config# vrf <vrf> snmp traps
vrouter running traps# authfail-check enabled true|false
```

**enabled**

Enable or disable authentication failures monitoring.

```
enabled true|false
```

**Default value**

true

## link-status-check

Monitor network interfaces being taken up or down, triggering a linkUp or linkDown notification as appropriate.

```
vrouter running config# vrf <vrf> snmp traps
vrouter running traps# link-status-check frequency FREQUENCY enabled true|false
```

### frequency

Check for network interfaces being taken up or down every <frequency> period.

```
frequency FREQUENCY
```

|                |  |
|----------------|--|
| FRE-<br>QUENCY | Value in seconds or optionnally suffixed by one of s (for seconds), m (for minutes), h (for hours), d (for days) or w (for weeks). |
|----------------|--|

#### Default value

60s

### enabled

Enable or disable link status monitoring.

```
enabled true|false
```

#### Default value

true

## process-check

Monitor the important processes of the system, triggering a notification when one of them is not alive.

```
vrouter running config# vrf <vrf> snmp traps
vrouter running traps# process-check frequency FREQUENCY enabled true|false
```

**frequency**

Check for network interfaces being taken up or down every <frequency> period.

```
frequency FREQUENCY
```

|                |  |
|----------------|--|
| FRE-<br>QUENCY | Value in seconds or optionnally suffixed by one of s (for seconds), m (for minutes), h (for hours), d (for days) or w (for weeks). |
|----------------|--|

**Default value**

2s

**enabled**

Enable or disable process monitoring.

```
enabled true|false
```

**Default value**

true

**disk-space-check**

Enables monitoring of all disks found on the system, using the specified (percentage) threshold.

```
vrouter running config# vrf <vrf> snmp traps
vrouter running traps# disk-space-check threshold <uint8> frequency FREQUENCY \
... enabled true|false
```

**threshold (mandatory)**

The minimum free disk space in percentage of the total space.

```
threshold <uint8>
```

## frequency

Check for free disk space every <frequency> period.

```
frequency FREQUENCY
```

|                |  |
|----------------|--|
| FRE-<br>QUENCY | Value in seconds or optionnally suffixed by one of s (for seconds), m (for minutes), h (for hours), d (for days) or w (for weeks). |
|----------------|--|

### Default value

5m

## enabled

Enable or disable disk space monitoring.

```
enabled true|false
```

### Default value

true

## load-check

Enables monitoring of the load average and trigger notifications if it goes above the specified thresholds.

```
vrouter running config# vrf <vrf> snmp traps
vrouter running traps# load-check threshold <uint16> enabled true|false
```

## threshold (mandatory)

The maximum system load average.

```
threshold <uint16>
```

### enabled

Enable or disable system load monitoring.

```
enabled true|false
```

### Default value

true

## 3.2.28 routing

### global

---

**Note:** requires a Turbo Router Network License.

---

Routing global configuration.

```
vrouter running config# routing
```

### ipv4-access-list

IPv4 access list.

```
vrouter running config# routing ipv4-access-list <string>
```

|          |                   |
|----------|-------------------|
| <string> | Access list name. |
|----------|-------------------|

### remark

Access list entry comment.

```
vrouter running config# routing ipv4-access-list <string>  
vrouter running ipv4-access-list <string># remark <string>
```

## seq

Specify access list to reject or accept.

```
vrouter running config# routing ipv4-access-list <string>
vrouter running ipv4-access-list <string># seq <uint16> \
... permit <permit> exact-match true|false \
... deny <deny> exact-match true|false
```

|          |                |
|----------|----------------|
| <uint16> | List sequence. |
|----------|----------------|

## permit

IPv4 access list deny rules.

```
permit <permit> exact-match true|false
```

## exact-match

Enable or disable exact match of the prefixes.

```
exact-match true|false
```

## deny

IPv4 access list deny rules.

```
deny <deny> exact-match true|false
```

## exact-match

Enable or disable exact match of the prefixes.

```
exact-match true|false
```

## ipv6-access-list

IPv6 access list.

```
vrouter running config# routing ipv6-access-list <string>
```

|          |                   |
|----------|-------------------|
| <string> | Access list name. |
|----------|-------------------|

## remark

Access list entry comment.

```
vrouter running config# routing ipv6-access-list <string>
vrouter running ipv6-access-list <string># remark <string>
```

## seq

Specify access list to reject or accept.

```
vrouter running config# routing ipv6-access-list <string>
vrouter running ipv6-access-list <string># seq <uint16> \
... permit <permit> exact-match true|false \
... deny <deny> exact-match true|false
```

|          |                       |
|----------|-----------------------|
| <uint16> | Access list sequence. |
|----------|-----------------------|

## permit

IPv6 access list deny rules.

```
permit <permit> exact-match true|false
```

## exact-match

Enable or disable exact match of the prefixes.

```
exact-match true|false
```

## deny

IPv6 access list deny rules.

```
deny <deny> exact-match true|false
```

## exact-match

Enable or disable exact match of the prefixes.

```
exact-match true|false
```

## logging

Logs configuration.

```
vrouter running config# routing logging
```

## enabled

Enable/Disable routing logs.

```
vrouter running config# routing logging  
vrouter running logging# enabled true|false
```

## Default value

true

## level

Set minimal logging level.

```
vrouter running config# routing logging  
vrouter running logging# level LEVEL
```

| LEVEL values | Description                       |
|--------------|-----------------------------------|
| emergency    | System is unusable.               |
| alert        | Action must be taken immediately. |
| critical     | Critical conditions.              |
| error        | Error conditions.                 |
| warning      | Warning conditions.               |
| notice       | Normal but significant condition. |
| info         | Informational messages.           |
| debug        | Debug-level messages.             |

**Default value**

error

**mpls**

MPLS logging configuration.

```
vrouter running config# routing logging mpls
```

**ldp**

Common LDP routers logging configuration.

```
vrouter running config# routing logging mpls ldp
```

**enabled**

Enable/disable MPLS LDP logging configuration.

```
vrouter running config# routing logging mpls ldp
vrouter running ldp# enabled true|false
```

**Default value**

true

## discovery-hello

Direction of discovery messages to log.

```
vrouter running config# routing logging mpls ldp
vrouter running ldp# discovery-hello DISCOVERY-HELLO
```

| DISCOVERY-HELLO values | Description            |
|------------------------|------------------------|
| send                   | Log sent messages.     |
| receive                | Log received messages. |
| both                   | Log all messages.      |

## errors

Log errors.

```
vrouter running config# routing logging mpls ldp
vrouter running ldp# errors true|false
```

### Default value

true

## events

Log event information.

```
vrouter running config# routing logging mpls ldp
vrouter running ldp# events true|false
```

### Default value

false

## labels

Log label allocation information.

```
vrouter running config# routing logging mpls ldp
vrouter running ldp# labels true|false
```

### Default value

false

**zebra**

Log zebra information.

```
vrouter running config# routing logging mpls ldp
vrouter running ldp# zebra true|false
```

**Default value**

false

**message**

Log LDP message information.

```
vrouter running config# routing logging mpls ldp message
```

**direction**

Direction of messages to log.

```
vrouter running config# routing logging mpls ldp message
vrouter running message# direction DIRECTION
```

| DIRECTION values | Description            |
|------------------|------------------------|
| send             | Log sent messages.     |
| receive          | Log received messages. |
| both             | Log all messages.      |

**Default value**

both

**detail**

Log message including periodic Keep Alives.

```
vrouter running config# routing logging mpls ldp message
vrouter running message# detail true|false
```

**Default value**

false

## bgp

---

**Note:** requires a Turbo Router Network License.

---

Common BGP routers logging configuration.

```
vrouter running config# routing logging bgp
```

### enabled

Enable/disable BGP logging configuration.

```
vrouter running config# routing logging bgp
vrouter running bgp# enabled true|false
```

### Default value

true

### allow-martians

Allow martian next hops.

```
vrouter running config# routing logging bgp
vrouter running bgp# allow-martians true|false
```

### Default value

false

### as-4bytes

Log AS > 65535 actions.

```
vrouter running config# routing logging bgp
vrouter running bgp# as-4bytes true|false
```

### Default value

false

## as-4bytes-segment

Log AS > 65535 aspath segment handling.

```
vrouter running config# routing logging bgp
vrouter running bgp# as-4bytes-segment true|false
```

### Default value

false

## bestpath

Log BGP bestpath info.

```
vrouter running config# routing logging bgp
vrouter running bgp# bestpath BESTPATH
```

| BESTPATH values | Description                            |
|-----------------|--|
| <A.B.C.D/M>     | An IPv4 prefix: address and CIDR mask. |
| <X:X::X:X/M>    | An IPv6 prefix: address and CIDR mask. |

## nexthop-tracking

Log BGP nexthop tracking.

```
vrouter running config# routing logging bgp
vrouter running bgp# nexthop-tracking true|false
```

### Default value

false

## flowspec

Enable flowspec debugging entries.

```
vrouter running config# routing logging bgp
vrouter running bgp# flowspec true|false
```

### Default value

false

## keepalives

Log keepalive messages to/from a specific neighbor or all.

```
vrouter running config# routing logging bgp
vrouter running bgp# keepalives KEEPALIVES
```

| KEEPALIVES values | Description                 |
|-------------------|-----------------------------|
| <A.B.C.D>         | An IPv4 address.            |
| <X:X::X:X>        | An IPv6 address.            |
| all               | Log all keepalive messages. |

## neighbor-events

Log neighbor event messages to/from a specific neighbor or all.

```
vrouter running config# routing logging bgp
vrouter running bgp# neighbor-events NEIGHBOR-EVENTS
```

| NEIGHBOR-EVENTS values | Description                      |
|------------------------|----------------------------------|
| <A.B.C.D>              | An IPv4 address.                 |
| <X:X::X:X>             | An IPv6 address.                 |
| all                    | Log all neighbor event messages. |

## update-groups

Log update messages (only when BGP is configured as a server).

```
vrouter running config# routing logging bgp
vrouter running bgp# update-groups true|false
```

### Default value

false

## zebra

Log zebra/BGP messages for a specific prefix or all.

```
vrouter running config# routing logging bgp
vrouter running bgp# zebra ZEBRA
```

| ZEBRA values | Description                             |
|--------------|---|
| <A.B.C.D/M>  | An IPv4 prefix: address and CIDR mask.  |
| <X:X::X:X/M> | An IPv6 prefix: address and CIDR mask.  |
| all          | Log all messages between Zebra and BGP. |

## rpki

Enable RPKI logging.

```
vrouter running config# routing logging bgp
vrouter running bgp# rpki true|false
```

### Default value

false

## pbr

Log policy base routing info.

```
vrouter running config# routing logging bgp pbr
```

## detail

Log policy base routing info with more details.

```
vrouter running config# routing logging bgp pbr
vrouter running pbr# detail true|false
```

### Default value

false

## updates

Log inbound and outbound update messages.

```
vrouter running config# routing logging bgp updates
```

## enabled

Enable/Disable log about inbound and outbound update messages.

```
vrouter running config# routing logging bgp updates
vrouter running updates# enabled true|false
```

## Default value

true

## in

Log inbound update messages from a specific neighbor or all.

```
vrouter running config# routing logging bgp updates
vrouter running updates# in IN
```

| IN values  | Description                                     |
|------------|---|
| <A.B.C.D>  | An IPv4 address.                                |
| <X:X::X:X> | An IPv6 address.                                |
| all        | Log inbound update messages from all neighbors. |

## Default value

all

## out

Log outbound update messages from a specific neighbor or all.

```
vrouter running config# routing logging bgp updates
vrouter running updates# out OUT
```

| OUT values | Description                                      |
|------------|--|
| <A.B.C.D>  | An IPv4 address.                                 |
| <X:X::X:X> | An IPv6 address.                                 |
| all        | Log outbound update messages from all neighbors. |

**Default value**

all

**prefix**

Log update messages to/from a specific network.

```
vrouters running config# routing logging bgp updates
vrouters running updates# prefix PREFIX
```

| PREFIX values | Description                            |
|---------------|--|
| <A.B.C.D/M>   | An IPv4 prefix: address and CIDR mask. |
| <X:X::X:X/M>  | An IPv6 prefix: address and CIDR mask. |

**vpn**

Log VPN routes.

```
vrouters running config# routing logging bgp vpn
```

**label**

Log VPN label.

```
vrouters running config# routing logging bgp vpn
vrouters running vpn# label true|false
```

**Default value**

false

## leak-vrf

Log leaks.

```
vrouter running config# routing logging bgp vpn
vrouter running vpn# leak-vrf LEAK-VRF
```

| LEAK-VRF values | Description               |
|-----------------|---------------------------|
| to              | Log leak to VRF from VPN. |
| from            | Log leak from VRF to VPN. |
| both            | Log all leaks.            |

## route-map-event

Log VPN route-map updates.

```
vrouter running config# routing logging bgp vpn
vrouter running vpn# route-map-event true|false
```

### Default value

false

## rip

**Note:** requires a Turbo Router Network License.

Common RIP routers logging configuration.

```
vrouter running config# routing logging rip
```

## enabled

Enable/disable router logging configuration.

```
vrouter running config# routing logging rip
vrouter running rip# enabled true|false
```

### Default value

true

**events**

Log router events.

```
vrouter running config# routing logging rip
vrouter running rip# events true|false
```

**Default value**

false

**packet**

Log router received/send packet info.

```
vrouter running config# routing logging rip
vrouter running rip# packet PACKET
```

| PACKET values | Description                    |
|---------------|--------------------------------|
| receive       | Log only received packet info. |
| send          | Log only sent packet info.     |
| both          | Log all packet info.           |

**Default value**

both

**zebra**

Log communication between the router and zebra.

```
vrouter running config# routing logging rip
vrouter running rip# zebra true|false
```

**Default value**

false

## ripng

---

**Note:** requires a Turbo Router Network License.

---

Common RIPng routers logging configuration.

```
vrouter running config# routing logging ripng
```

## enabled

Enable/disable router logging configuration.

```
vrouter running config# routing logging ripng
vrouter running ripng# enabled true|false
```

## Default value

true

## events

Log router events.

```
vrouter running config# routing logging ripng
vrouter running ripng# events true|false
```

## Default value

false

## packet

Log router received/send packet info.

```
vrouter running config# routing logging ripng
vrouter running ripng# packet PACKET
```

| PACKET values | Description                    |
|---------------|--------------------------------|
| receive       | Log only received packet info. |
| send          | Log only sent packet info.     |
| both          | Log all packet info.           |

**Default value**

both

**zebra**

Log communication between the router and zebra.

```
vrouter running config# routing logging ripng
vrouter running ripng# zebra true|false
```

**Default value**

false

**ospf**

---

**Note:** requires a Turbo Router Network License.

---

Common OSPF routers logging configuration.

```
vrouter running config# routing logging ospf
```

**enabled**

Enable/disable OSPF logging configuration.

```
vrouter running config# routing logging ospf
vrouter running ospf# enabled true|false
```

**Default value**

true

**events**

Log OSPF event information.

```
vrouter running config# routing logging ospf
vrouter running ospf# events true|false
```

**Default value**

false

## ism

Log OSPF Interface State Machine information.

```
vrouter running config# routing logging ospf
vrouter running ospf# ism ISM
```

| ISM values | Description                 |
|------------|-----------------------------|
| events     | Log ISM Event Information.  |
| status     | Log ISM Status Information. |
| timers     | Log ISM Timer Information.  |
| all        | Log all ISM Information.    |

## lsa

Log OSPF Link State Advertisement information.

```
vrouter running config# routing logging ospf
vrouter running ospf# lsa LSA
```

| LSA values | Description                   |
|------------|-------------------------------|
| flooding   | Log LSA flooding Information. |
| generate   | Log LSA generate Information. |
| install    | Log LSA install Information.  |
| refresh    | Log LSA refresh Information.  |
| all        | Log all LSA Information.      |

## nsm

Log OSPF Neighbor State Machine information.

```
vrouter running config# routing logging ospf
vrouter running ospf# nsm NSM
```

| NSM values | Description                 |
|------------|-----------------------------|
| events     | Log NSM Event Information.  |
| status     | Log NSM Status Information. |
| timers     | Log NSM Timer Information.  |
| all        | Log all NSM Information.    |

**nssa**

Log OSPF nssa information.

```
vrouter running config# routing logging ospf
vrouter running ospf# nssa true|false
```

**Default value**

false

**zebra**

Log zebra information.

```
vrouter running config# routing logging ospf
vrouter running ospf# zebra ZEBRA
```

| ZEBRA values | Description                                       |
|--------------|---|
| interface    | Log zebra interface information.                  |
| redistribute | Log zebra redistribute information.               |
| all          | Log zebra interface and redistribute information. |

**message**

Log OSPF message information.

```
vrouter running config# routing logging ospf message <message>
```

| <message> values | Description                             |
|------------------|---|
| dd               | Log Database Description messages.      |
| hello            | Log Hello messages.                     |
| ls-ack           | Log Link State Acknowledgment messages. |
| ls-request       | Log Link State Request messages.        |
| ls-update        | Log Link State Update messages.         |
| all              | Log all messages (whatever its type).   |

### direction

Direction of messages to log.

```
vrouter running config# routing logging ospf message <message>  
vrouter running message <message># direction DIRECTION
```

| DIRECTION values | Description            |
|------------------|------------------------|
| send             | Log sent messages.     |
| receive          | Log received messages. |
| both             | Log all messages.      |

#### Default value

both

### detail

Log message details.

```
vrouter running config# routing logging ospf message <message>  
vrouter running message <message># detail true|false
```

#### Default value

false

### ospf6

---

**Note:** requires a Turbo Router Network License.

---

Common OSPF6 routers logging configuration.

```
vrouter running config# routing logging ospf6
```

## enabled

Enable/Disable OSPF6 logging configuration.

```
vrouter running config# routing logging ospf6
vrouter running ospf6# enabled true|false
```

### Default value

true

## abr

Log ABR information.

```
vrouter running config# routing logging ospf6
vrouter running ospf6# abr true|false
```

### Default value

false

## asbr

Log ASBR information.

```
vrouter running config# routing logging ospf6
vrouter running ospf6# asbr true|false
```

### Default value

false

## events

Log events.

```
vrouter running config# routing logging ospf6
vrouter running ospf6# events true|false
```

### Default value

false

## flooding

Log flooding information.

```
vrouter running config# routing logging ospf6
vrouter running ospf6# flooding true|false
```

### Default value

false

## interface

Log interface information.

```
vrouter running config# routing logging ospf6
vrouter running ospf6# interface true|false
```

### Default value

false

## neighbor

Log neighbor information.

```
vrouter running config# routing logging ospf6
vrouter running ospf6# neighbor NEIGHBOR
```

| NEIGHBOR values | Description                     |
|-----------------|---------------------------------|
| events          | Log neighbor event information. |
| state           | Log neighbor state information. |
| all             | Log all neighbor information.   |

## route

Log route information.

```
vrouter running config# routing logging ospf6
vrouter running ospf6# route ROUTE
```

| ROUTE values | Description                       |
|--------------|-----------------------------------|
| inter-area   | Log inter area route calculation. |
| intra-area   | Log intra area route calculation. |
| memory       | Log route memory use..            |
| table        | Log route table calculation.      |
| all          | Log all route information.        |

## spf

Log SPF calculation.

```
vrouter running config# routing logging ospf6
vrouter running ospf6# spf SPF
```

| SPF values | Description                                 |
|------------|---|
| database   | Log number of LSAs at SPF calculation time. |
| process    | Log detailed SPF process.                   |
| time       | Measure time taken by SPF calculation.      |
| all        | Log all SPF messages.                       |

## zebra

Log messages between OSPF router and zebra.

```
vrouter running config# routing logging ospf6
vrouter running ospf6# zebra ZEBRA
```

| ZEBRA values | Description                       |
|--------------|-----------------------------------|
| send         | Log messages sent to zebra.       |
| receive      | Log messages received from zebra. |
| both         | Log messages to/from zebra.       |

## border-routers

Log border routers information.

```
vrouter running config# routing logging ospf6 border-routers
```

## summary

Log border router information in a specific area.

```
vrouter running config# routing logging ospf6 border-routers
vrouter running border-routers# summary true|false
```

### Default value

false

## area-id

Log border router information in a specific area.

```
vrouter running config# routing logging ospf6 border-routers
vrouter running border-routers# area-id AREA-ID
```

|         |                  |
|---------|------------------|
| AREA-ID | An IPv4 address. |
|---------|------------------|

## router-id

Log information from a specific border router.

```
vrouter running config# routing logging ospf6 border-routers
vrouter running border-routers# router-id ROUTER-ID
```

|           |                  |
|-----------|------------------|
| ROUTER-ID | An IPv4 address. |
|-----------|------------------|

## lsa

Configure Link State Advertisements logging information.

```
vrouter running config# routing logging ospf6
vrouter running ospf6# lsa <lsa> level LEVEL
```

| <lsa> values | Description                 |
|--------------|-----------------------------|
| as-external  | Log as-external LSAs.       |
| inter-prefix | Log inter area prefix LSAs. |
| inter-router | LOG inter router LSAs.      |
| intra-prefix | LOG intra area prefix LSAs. |
| link         | LOG link LSAs.              |
| network      | LOG network LSAs.           |
| router       | LOG router LSAs.            |
| all          | LOG all LSA information.    |

**level**

LSA log level.

```
level LEVEL
```

| LEVEL values | Description                     |
|--------------|---------------------------------|
| examine      | Dump LSAs.                      |
| flooding     | Log LSA’s internal information. |
| originate    | Log details of LSAs.            |
| all          | Log all information about LSAs. |

**Default value**

all

**message**

Log OSPF message information.

```
vrouter running config# routing logging ospf6
vrouter running ospf6# message <message> direction DIRECTION
```

| <message> values | Description                             |
|------------------|---|
| dd               | Log Database Description messages.      |
| hello            | Log Hello messages.                     |
| ls-ack           | Log Link State Acknowledgment messages. |
| ls-request       | Log Link State Request messages.        |
| ls-update        | Log Link State Update messages.         |
| all              | Log all messages.                       |

## direction

Direction of messages to log.

```
direction DIRECTION
```

| DIRECTION values | Description            |
|------------------|------------------------|
| send             | Log sent messages.     |
| receive          | Log received messages. |
| both             | Log all messages.      |

### Default value

both

## ipv4-prefix-list

IPv4 prefix list.

```
vrouter running config# routing ipv4-prefix-list <string>
```

|          |                   |
|----------|-------------------|
| <string> | Prefix list name. |
|----------|-------------------|

## seq

Prefix list sequence.

```
vrouter running config# routing ipv4-prefix-list <string>
vrouter running ipv4-prefix-list <string># seq <uint32> address ADDRESS policy POLICY \
... ge <uint8> le <uint8>
```

|          |                  |
|----------|------------------|
| <uint32> | Sequence number. |
|----------|------------------|

## address

Prefix to match (any if not set).

```
address ADDRESS
```

|         |  |
|---------|--|
| ADDRESS | An IPv4 prefix: address and CIDR mask. |
|---------|--|

**policy (mandatory)**

Prefix list policy.

```
policy POLICY
```

| POLICY values | Description                 |
|---------------|-----------------------------|
| deny          | Specify packets to reject.  |
| permit        | Specify packets to forward. |

**ge**

Minimum prefix length to be matched.

```
ge <uint8>
```

**le**

Maximum prefix length to be matched.

```
le <uint8>
```

**ipv6-prefix-list**

IPv6 prefix list.

```
vrouter running config# routing ipv6-prefix-list <string>
```

|          |                   |
|----------|-------------------|
| <string> | Prefix list name. |
|----------|-------------------|

**seq**

Prefix list sequence.

```
vrouter running config# routing ipv6-prefix-list <string>
vrouter running ipv6-prefix-list <string># seq <uint32> address ADDRESS policy POLICY \
... ge <uint8> le <uint8>
```

|          |                  |
|----------|------------------|
| <uint32> | Sequence number. |
|----------|------------------|

### address

Prefix to match (any if not set).

```
address ADDRESS
```

|         |  |
|---------|--|
| ADDRESS | An IPv6 prefix: address and CIDR mask. |
|---------|--|

### policy (mandatory)

Prefix list policy.

```
policy POLICY
```

| POLICY values | Description                 |
|---------------|-----------------------------|
| deny          | Specify packets to reject.  |
| permit        | Specify packets to forward. |

### ge

Minimum prefix length to be matched.

```
ge <uint8>
```

### le

Maximum prefix length to be matched.

```
le <uint8>
```

### route-map

Route map list.

```
vrouter running config# routing route-map <string>
```

|          |                 |
|----------|-----------------|
| <string> | Route map name. |
|----------|-----------------|

## seq

Route map sequence.

```
vrouter running config# routing route-map <string> seq <uint16>
```

|          |                  |
|----------|------------------|
| <uint16> | Sequence number. |
|----------|------------------|

## policy (mandatory)

Matching policy.

```
vrouter running config# routing route-map <string> seq <uint16>
vrouter running seq <uint16># policy POLICY
```

| POLICY values | Description                       |
|---------------|-----------------------------------|
| deny          | Route map denies set operations.  |
| permit        | Route map permits set operations. |

## description

Route-map description.

```
vrouter running config# routing route-map <string> seq <uint16>
vrouter running seq <uint16># description <string>
```

## call

Jump to another Route-Map after match+set.

```
vrouter running config# routing route-map <string> seq <uint16>
vrouter running seq <uint16># call <string>
```

## on-match

Exit policy on matches.

```
vrouter running config# routing route-map <string> seq <uint16>
vrouter running seq <uint16># on-match ON-MATCH
```

| ON-MATCH values | Description     |
|-----------------|-----------------|
| <uint16>        | No description. |
| next            | Next clause.    |

## match

Match values from routing table.

```
vrouter running config# routing route-map <string> seq <uint16> match
```

## as-path

Match BGP AS path list.

```
vrouter running config# routing route-map <string> seq <uint16> match
vrouter running match# as-path <string>
```

## interface

Match first hop interface of route.

```
vrouter running config# routing route-map <string> seq <uint16> match
vrouter running match# interface INTERFACE
```

|           |                    |
|-----------|--------------------|
| INTERFACE | An interface name. |
|-----------|--------------------|

## local-preference

Match local-preference metric value.

```
vrouter running config# routing route-map <string> seq <uint16> match
vrouter running match# local-preference <uint32>
```

## mac-address

Match MAC Access-list name.

```
vrouter running config# routing route-map <string> seq <uint16> match
vrouter running match# mac-address <string>
```

## metric

Match metric value.

```
vrouter running config# routing route-map <string> seq <uint16> match
vrouter running match# metric <uint32>
```

## origin

BGP origin code.

```
vrouter running config# routing route-map <string> seq <uint16> match
vrouter running match# origin ORIGIN
```

| ORIGIN values | Description       |
|---------------|-------------------|
| egp           | Remote EGP.       |
| igp           | Local IGP.        |
| incomplete    | Unknown heritage. |

## peer

Match peer address.

```
vrouter running config# routing route-map <string> seq <uint16> match
vrouter running match# peer PEER
```

| PEER values | Description                     |
|-------------|---------------------------------|
| local       | Static or redistributed routes. |
| <A.B.C.D>   | An IPv4 address.                |
| <X:X::X:X>  | An IPv6 address.                |
| <ifname>    | An interface name.              |

## probability

Match portion of routes defined by percentage value.

```
vrouter running config# routing route-map <string> seq <uint16> match
vrouter running match# probability <uint8>
```

## source-instance

Match the protocol's instance number.

```
vrouter running config# routing route-map <string> seq <uint16> match
vrouter running match# source-instance <uint8>
```

## source-protocol

Match protocol via which the route was learnt.

```
vrouter running config# routing route-map <string> seq <uint16> match
vrouter running match# source-protocol SOURCE-PROTOCOL
```

| SOURCE-PROTOCOL values | Description                          |
|------------------------|--------------------------------------|
| babel                  | BABEL protocol.                      |
| bgp                    | BGP protocol.                        |
| connected              | Routes from directly connected peer. |
| eigrp                  | EIGRP protocol.                      |
| isis                   | ISIS protocol.                       |
| kernel                 | Routes from kernel.                  |
| nhrp                   | NHRP protocol.                       |
| ospf                   | OSPF protocol.                       |
| ospf6                  | OSPF6 protocol.                      |
| pim                    | PIM protocol.                        |
| rip                    | RIP protocol.                        |
| ripng                  | RIPNG protocol.                      |
| sharp                  | SHARP process.                       |
| static                 | Statically configured routes.        |
| system                 | Routes from system configuration.    |

## tag

Match tag of route.

```
vrouter running config# routing route-map <string> seq <uint16> match
vrouter running match# tag <uint32>
```

## extcommunity

Match BGP/VPN extended community list.

```
vrouter running config# routing route-map <string> seq <uint16> match
vrouter running match# extcommunity <leafref>
```

## rpki

RPKI specific settings.

```
vrouter running config# routing route-map <string> seq <uint16> match
vrouter running match# rpki RPKI
```

| RPKI values | Description        |
|-------------|--------------------|
| valid       | Valid prefix.      |
| invalid     | Invalid prefix.    |
| notfound    | Prerfix not found. |

## evpn

Ethernet Virtual Private Network.

```
vrouter running config# routing route-map <string> seq <uint16> match evpn
```

## default-route

If true, mark as default EVPN type-5 route.

```
vrouter running config# routing route-map <string> seq <uint16> match evpn
vrouter running evpn# default-route true|false
```

## route-type

Match route type.

```
vrouter running config# routing route-map <string> seq <uint16> match evpn
vrouter running evpn# route-type ROUTE-TYPE
```

| ROUTE-TYPE values | Description   |
|-------------------|---------------|
| macip             | Mac-ip route. |
| multicast         | IMET route.   |
| prefix            | Prefix route. |

## vni

VNI ID.

```
vrouter running config# routing route-map <string> seq <uint16> match evpn
vrouter running evpn# vni <uint32>
```

### route-distinguisher

Route distinguisher.

```
vrouter running config# routing route-map <string> seq <uint16> match evpn
vrouter running evpn# route-distinguisher ROUTE-DISTINGUISHER
```

| ROUTE-DISTINGUISHER values | Description   |
|----------------------------|---|
| <string>                   | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |
| <string>                   | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |

### ip

IP information.

```
vrouter running config# routing route-map <string> seq <uint16> match ip
```

### address

Match address of route.

```
vrouter running config# routing route-map <string> seq <uint16> match ip address
```

### access-list

Matches the specified access list.

```
vrouter running config# routing route-map <string> seq <uint16> match ip address
vrouter running address# access-list ACCESS-LIST
```

| ACCESS-LIST values | Description     |
|--------------------|-----------------|
| <uint16>           | No description. |
| <string>           | No description. |

**prefix-list**

Matches the specified prefix list.

```
vrouter running config# routing route-map <string> seq <uint16> match ip address
vrouter running address# prefix-list <string>
```

**prefix-len**

Matches the specified prefix length.

```
vrouter running config# routing route-map <string> seq <uint16> match ip address
vrouter running address# prefix-len <uint8>
```

**next-hop**

Match next-hop address of route.

```
vrouter running config# routing route-map <string> seq <uint16> match ip next-hop
```

**access-list**

Matches the specified access list.

```
vrouter running config# routing route-map <string> seq <uint16> match ip next-hop
vrouter running next-hop# access-list ACCESS-LIST
```

| ACCESS-LIST values | Description     |
|--------------------|-----------------|
| <uint16>           | No description. |
| <string>           | No description. |

**prefix-list**

Matches the specified prefix list.

```
vrouter running config# routing route-map <string> seq <uint16> match ip next-hop
vrouter running next-hop# prefix-list <string>
```

## prefix-len

Matches the specified prefix length.

```
vrouter running config# routing route-map <string> seq <uint16> match ip next-hop
vrouter running next-hop# prefix-len <uint8>
```

## route-source

Match advertising source address of route.

```
vrouter running config# routing route-map <string> seq <uint16> match ip route-source
```

## access-list

Matches the specified access list.

```
vrouter running config# routing route-map <string> seq <uint16> match ip route-source
vrouter running route-source# access-list ACCESS-LIST
```

| ACCESS-LIST values | Description     |
|--------------------|-----------------|
| <uint16>           | No description. |
| <string>           | No description. |

## prefix-list

Matches the specified prefix list.

```
vrouter running config# routing route-map <string> seq <uint16> match ip route-source
vrouter running route-source# prefix-list <string>
```

## ipv6

IPv6 information.

```
vrouter running config# routing route-map <string> seq <uint16> match ipv6
```

## address

Match IPv6 address of route.

```
vrouter running config# routing route-map <string> seq <uint16> match ipv6 address
```

## access-list

Matches the specified access list.

```
vrouter running config# routing route-map <string> seq <uint16> match ipv6 address  
vrouter running address# access-list <string>
```

## prefix-list

Matches the specified prefix list.

```
vrouter running config# routing route-map <string> seq <uint16> match ipv6 address  
vrouter running address# prefix-list <string>
```

## prefix-len

Matches the specified prefix length.

```
vrouter running config# routing route-map <string> seq <uint16> match ipv6 address  
vrouter running address# prefix-len <uint8>
```

## next-hop

Match IPv6 next-hop address of route.

```
vrouter running config# routing route-map <string> seq <uint16> match ipv6 next-hop
```

## address

IPv6 address of next hop.

```
vrouter running config# routing route-map <string> seq <uint16> match ipv6 next-hop  
vrouter running next-hop# address ADDRESS
```

|         |                  |
|---------|------------------|
| ADDRESS | An IPv6 address. |
|---------|------------------|

## community

Match BGP community list.

```
vrouter running config# routing route-map <string> seq <uint16> match  
vrouter running match# community id <leafref> exact-match true|false
```

## id (mandatory)

Community-list number or name.

```
id <leafref>
```

## exact-match

If true, do exact matching of communities.

```
exact-match true|false
```

## set

Set values in destination routing protocol.

```
vrouter running config# routing route-map <string> seq <uint16> set
```

## table

Export route to non-main table.

```
vrouter running config# routing route-map <string> seq <uint16> set  
vrouter running set# table <uint32>
```

## atomic-aggregate

Enable or disable BGP atomic aggregate attribute.

```
vrouter running config# routing route-map <string> seq <uint16> set  
vrouter running set# atomic-aggregate true|false
```

## label-index

Label index value.

```
vrouter running config# routing route-map <string> seq <uint16> set  
vrouter running set# label-index <uint32>
```

## local-preference

BGP local preference path attribute.

```
vrouter running config# routing route-map <string> seq <uint16> set  
vrouter running set# local-preference <uint32>
```

## metric

Metric value for destination routing protocol.

```
vrouter running config# routing route-map <string> seq <uint16> set  
vrouter running set# metric METRIC
```

| METRIC values   | Description               |
|-----------------|---------------------------|
| <uint32>        | No description.           |
| add-metric      | Add metric.               |
| add-rtt         | Add round trip time.      |
| subtract-metric | Subtract metric.          |
| subtract-rtt    | Subtract round trip time. |
| rtt             | Assign round trip time.   |

### metric-type

Type of metric.

```
vrouter running config# routing route-map <string> seq <uint16> set
vrouter running set# metric-type METRIC-TYPE
```

| METRIC-TYPE values | Description                   |
|--------------------|-------------------------------|
| type-1             | OSPF6 external type 1 metric. |
| type-2             | OSPF6 external type 2 metric. |

### origin

BGP origin code.

```
vrouter running config# routing route-map <string> seq <uint16> set
vrouter running set# origin ORIGIN
```

| ORIGIN values | Description       |
|---------------|-------------------|
| egp           | Remote EGP.       |
| igp           | Local IGP.        |
| incomplete    | Unknown heritage. |

### originator-id

BGP originator ID attribute.

```
vrouter running config# routing route-map <string> seq <uint16> set
vrouter running set# originator-id ORIGINATOR-ID
```

|               |                  |
|---------------|------------------|
| ORIGINATOR-ID | An IPv4 address. |
|---------------|------------------|

**src**

Src address for route.

```
vrouter running config# routing route-map <string> seq <uint16> set
vrouter running set# src SRC
```

| SRC values | Description      |
|------------|------------------|
| <A.B.C.D>  | An IPv4 address. |
| <X:X::X:X> | An IPv6 address. |

**tag**

Tag value for routing protocol.

```
vrouter running config# routing route-map <string> seq <uint16> set
vrouter running set# tag <uint32>
```

**weight**

BGP weight for routing table.

```
vrouter running config# routing route-map <string> seq <uint16> set
vrouter running set# weight <uint32>
```

**comm-list-delete**

Set BGP community list (for deletion).

```
vrouter running config# routing route-map <string> seq <uint16> set
vrouter running set# comm-list-delete <leafref>
```

**aggregator**

BGP aggregator attribute.

```
vrouter running config# routing route-map <string> seq <uint16> set
vrouter running set# aggregator as <uint32> address ADDRESS
```

### as (mandatory)

AS number of BGP aggregator.

```
as <uint32>
```

### address (mandatory)

IP address of aggregator.

```
address ADDRESS
```

|         |                  |
|---------|------------------|
| ADDRESS | An IPv4 address. |
|---------|------------------|

### as-path

Transform BGP AS-path attribute.

```
vrouter running config# routing route-map <string> seq <uint16> set as-path
```

### exclude

AS numbers to exclude from the as-path.

```
vrouter running config# routing route-map <string> seq <uint16> set as-path  
vrouter running as-path# exclude <uint32>
```

### prepend

Prepend to the as-path.

```
vrouter running config# routing route-map <string> seq <uint16> set as-path prepend
```

## last-as

Use the peer's AS-number; number of times to insert.

```
vrouter running config# routing route-map <string> seq <uint16> set as-path prepend
vrouter running prepend# last-as <uint8>
```

## asn

AS number to prepend to the as-path list.

```
vrouter running config# routing route-map <string> seq <uint16> set as-path prepend
↳asn <uint8>
```

|         |                         |
|---------|-------------------------|
| <uint8> | Order of the AS number. |
|---------|-------------------------|

## <uint32> (mandatory)

AS number.

```
vrouter running config# routing route-map <string> seq <uint16> set as-path prepend
↳asn <uint8>
vrouter running asn <uint8># <uint32>
```

## ip

IP information.

```
vrouter running config# routing route-map <string> seq <uint16> set ip
```

## next-hop

Next hop address.

```
vrouter running config# routing route-map <string> seq <uint16> set ip
vrouter running ip# next-hop NEXT-HOP
```

| NEXT-HOP values | Description                             |
|-----------------|---|
| <A.B.C.D>       | An IPv4 address.                        |
| peer-address    | Use peer address (for BGP only).        |
| unchanged       | Don't modify existing Next hop address. |

## ipv4

IPv4 information.

```
vrouter running config# routing route-map <string> seq <uint16> set ipv4
```

## vpn

VPN information.

```
vrouter running config# routing route-map <string> seq <uint16> set ipv4 vpn
```

## next-hop

VPN next-hop address.

```
vrouter running config# routing route-map <string> seq <uint16> set ipv4 vpn
vrouter running vpn# next-hop NEXT-HOP
```

|          |                  |
|----------|------------------|
| NEXT-HOP | An IPv4 address. |
|----------|------------------|

## ipv6

IPv6 information.

```
vrouter running config# routing route-map <string> seq <uint16> set ipv6
```

## next-hop

IPv6 next hop address.

```
vrouter running config# routing route-map <string> seq <uint16> set ipv6 next-hop
```

## global

IPv6 global address.

```
vrouter running config# routing route-map <string> seq <uint16> set ipv6 next-hop
vrouter running next-hop# global GLOBAL
```

|        |                  |
|--------|------------------|
| GLOBAL | An IPv6 address. |
|--------|------------------|

## local

IPv6 local address.

```
vrouter running config# routing route-map <string> seq <uint16> set ipv6 next-hop
vrouter running next-hop# local LOCAL
```

|       |  |
|-------|--|
| LOCAL | An IPv6 link-local address which is in the range of fe80::/10. |
|-------|--|

## peer-address

If true, use peer address (for BGP only).

```
vrouter running config# routing route-map <string> seq <uint16> set ipv6 next-hop
vrouter running next-hop# peer-address true|false
```

## prefer-global

If true, prefer global over link-local if both exist.

```
vrouter running config# routing route-map <string> seq <uint16> set ipv6 next-hop
vrouter running next-hop# prefer-global true|false
```

## vpn

VPN information.

```
vrouter running config# routing route-map <string> seq <uint16> set ipv6 vpn
```

## next-hop

VPN next-hop address.

```
vrouter running config# routing route-map <string> seq <uint16> set ipv6 vpn
vrouter running vpn# next-hop NEXT-HOP
```

|          |                  |
|----------|------------------|
| NEXT-HOP | An IPv6 address. |
|----------|------------------|

## community

BGP community attribute.

```
vrouter running config# routing route-map <string> seq <uint16> set community
```

## none

Set community to none.

```
vrouter running config# routing route-map <string> seq <uint16> set community
vrouter running community# none
```

## attribute

BGP community attribute.

```
vrouter running config# routing route-map <string> seq <uint16> set community
vrouter running community# attribute ATTRIBUTE
```

| ATTRIBUTE values           | Description                 |
|----------------------------|-----------------------------|
| local-AS                   | Local AS.                   |
| no-advertise               | Do not advertise.           |
| no-export                  | Do not export.              |
| internet                   | Internet.                   |
| graceful-shutdown          | Graceful-shutdown.          |
| accept-own                 | Accept-own.                 |
| route-filter-translated-v4 | Route-filter-translated-v4. |
| route-filter-v4            | Route-filter-v4.            |
| route-filter-translated-v6 | Route-filter-translated-v6. |
| route-filter-v6            | Route-filter-v6.            |
| llgr-stale                 | Llgr-stale.                 |
| no-llgr                    | No-llgr.                    |
| accept-own-nextthop        | Accept-own-nextthop.        |
| blackhole                  | Blackhole.                  |
| no-peer                    | No-peer.                    |
| <string>                   | Community attribute.        |
| additive                   | Additive.                   |

### extcommunity

BGP extended community attribute.

```
vrouter running config# routing route-map <string> seq <uint16> set extcommunity
```

### rt

Route Target extended community.

```
vrouter running config# routing route-map <string> seq <uint16> set extcommunity
vrouter running extcommunity# rt RT
```

| RT values | Description                                     |
|-----------|---|
| <string>  | Extended communities or route-target attribute. |
| <string>  | Extended communities or route-target attribute. |
| <string>  | Extended communities or route-target attribute. |

**soo**

Site-of-Origin extended community.

```
vrouter running config# routing route-map <string> seq <uint16> set extcommunity
vrouter running extcommunity# soo S00
```

| S00 values | Description                                     |
|------------|---|
| <string>   | Extended communities or route-target attribute. |
| <string>   | Extended communities or route-target attribute. |
| <string>   | Extended communities or route-target attribute. |

**bgp**


---

**Note:** requires a Turbo Router Network License.

---

Common BGP routers configuration.

```
vrouter running config# routing bgp
```

**route-map-delay**

Time in secs to wait before processing route-map changes.

```
vrouter running config# routing bgp
vrouter running bgp# route-map-delay <uint16>
```

**Default value**

5

**rpki-ssh-key (state only)**

Available SSH key pairs.

```
vrouter> show state routing bgp rpki-ssh-key
```

### community-list

Add a community list entry.

```
vrouter running config# routing bgp community-list <community-list>
```

| <community-list> values | Description |
|-------------------------|-------------|
| <uint8>                 | List name.  |
| <string>                | List name.  |

### policy

Specify communities to reject or accept.

```
vrouter running config# routing bgp community-list <community-list>
vrouter running community-list <community-list># policy <uint16> POLICY COMMUNITY
```

|          |   |
|----------|---|
| <uint16> | Priority of the policy. Lesser is the value, greater is the priority. |
|----------|---|

### POLICY (mandatory)

Policy to apply to the specified communities.

```
POLICY
```

| POLICY values | Description                             |
|---------------|---|
| deny          | Specified communities will be rejected. |
| permit        | Specified communities will be accepted. |

### COMMUNITY

Communities on which the policy should be applied.

```
COMMUNITY
```

| COMMUNITY values           | Description                 |
|----------------------------|-----------------------------|
| local-AS                   | Local AS.                   |
| no-advertise               | Do not advertise.           |
| no-export                  | Do not export.              |
| internet                   | Internet.                   |
| graceful-shutdown          | Graceful-shutdown.          |
| accept-own                 | Accept-own.                 |
| route-filter-translated-v4 | Route-filter-translated-v4. |
| route-filter-v4            | Route-filter-v4.            |
| route-filter-translated-v6 | Route-filter-translated-v6. |
| route-filter-v6            | Route-filter-v6.            |
| llgr-stale                 | Llgr-stale.                 |
| no-llgr                    | No-llgr.                    |
| accept-own-nexthop         | Accept-own-nexthop.         |
| blackhole                  | Blackhole.                  |
| no-peer                    | No-peer.                    |
| <string>                   | Community attribute.        |

### community-list-expanded

Add an expanded community list entry.

```
vrouter running config# routing bgp community-list-expanded <community-list-expanded>
```

| <community-list-expanded> values | Description |
|----------------------------------|-------------|
| <uint16>                         | List name.  |
| <string>                         | List name.  |

### policy

Specify communities to reject or accept.

```
vrouter running config# routing bgp community-list-expanded <community-list-expanded>
vrouter running community-list-expanded <community-list-expanded># policy <uint16> \
... POLICY COMMUNITY
```

|          |   |
|----------|---|
| <uint16> | Priority of the policy. Lesser is the value, greater is the priority. |
|----------|---|

### POLICY (mandatory)

Policy to apply to the specified communities.

POLICY

| POLICY values | Description                             |
|---------------|---|
| deny          | Specified communities will be rejected. |
| permit        | Specified communities will be accepted. |

### COMMUNITY

Communities on which the policy should be applied.

COMMUNITY

| COMMUNITY values           | Description                 |
|----------------------------|-----------------------------|
| local-AS                   | Local AS.                   |
| no-advertise               | Do not advertise.           |
| no-export                  | Do not export.              |
| internet                   | Internet.                   |
| graceful-shutdown          | Graceful-shutdown.          |
| accept-own                 | Accept-own.                 |
| route-filter-translated-v4 | Route-filter-translated-v4. |
| route-filter-v4            | Route-filter-v4.            |
| route-filter-translated-v6 | Route-filter-translated-v6. |
| route-filter-v6            | Route-filter-v6.            |
| llgr-stale                 | Llgr-stale.                 |
| no-llgr                    | No-llgr.                    |
| accept-own-nexthop         | Accept-own-nexthop.         |
| blackhole                  | Blackhole.                  |
| no-peer                    | No-peer.                    |
| <string>                   | Community attribute.        |

### extcommunity-list

Add an extended community list entry.

```
vrouter running config# routing bgp extcommunity-list <extcommunity-list>
```

| <extcommunity-list> values | Description |
|----------------------------|-------------|
| <uint8>                    | List name.  |
| <string>                   | List name.  |

### policy

Specify extended communities to reject or accept.

```
vrouter running config# routing bgp extcommunity-list <extcommunity-list>
vrouter running extcommunity-list <extcommunity-list># policy <uint16> POLICY \
... rt RT soo S00
```

|          |   |
|----------|---|
| <uint16> | Priority of the policy. Lesser is the value, greater is the priority. |
|----------|---|

### POLICY (mandatory)

Policy to apply to the specified extcommunities.

```
POLICY
```

| POLICY values | Description                              |
|---------------|--|
| deny          | Specified extcommunities will be reject. |
| permit        | Specified extcommunities will be accept. |

### rt

Extended community route target to reject.

```
rt RT
```

| RT values | Description                                     |
|-----------|---|
| <string>  | Extended communities or route-target attribute. |
| <string>  | Extended communities or route-target attribute. |
| <string>  | Extended communities or route-target attribute. |

### soo

Extended community site of origin to reject.

```
soo S00
```

| S00 values | Description                                     |
|------------|---|
| <string>   | Extended communities or route-target attribute. |
| <string>   | Extended communities or route-target attribute. |
| <string>   | Extended communities or route-target attribute. |

### extcommunity-list-expanded

Add an expanded extended community list entry.

```
vrouter running config# routing bgp extcommunity-list-expanded <extcommunity-list-
↳expanded>
```

| <extcommunity-list-expanded> values | Description |
|-------------------------------------|-------------|
| <uint16>                            | List name.  |
| <string>                            | List name.  |

### policy

Specify extended communities to reject or accept.

```
vrouter running config# routing bgp extcommunity-list-expanded <extcommunity-list-
↳expanded>
vrouter running extcommunity-list-expanded <extcommunity-list-expanded># policy
↳<uint16> \
... POLICY <string>
```

|          |   |
|----------|---|
| <uint16> | Priority of the policy. Lesser is the value, greater is the priority. |
|----------|---|

### POLICY (mandatory)

Policy to apply to the specified extcommunities.

POLICY

| POLICY values | Description                              |
|---------------|--|
| deny          | Specified extcommunities will be reject. |
| permit        | Specified extcommunities will be accept. |

#### <string>

Communities on which the policy should be applied.

<string>

### as-path-access-list

BGP autonomous system path filter.

**vrouter running config#** routing bgp as-path-access-list <string>

|          |                   |
|----------|-------------------|
| <string> | Access list name. |
|----------|-------------------|

### policy

Specify AS path access list to reject or accept.

**vrouter running config#** routing bgp as-path-access-list <string>  
**vrouter running as-path-access-list <string>#** policy <uint16> POLICY <string>

|          |   |
|----------|---|
| <uint16> | Priority of the policy. Lesser is the value, greater is the priority. |
|----------|---|

## POLICY (mandatory)

Policy to apply to the specified regular expression that match AS paths.

```
POLICY
```

| POLICY values | Description                             |
|---------------|---|
| deny          | Specified access list will be rejected. |
| permit        | Specified access list will be accepted. |

### <string>

Regular expression to match the BGP AS paths on which the policy should be applied.

```
<string>
```

### static

Static routes.

```
vrouter running config# vrf <vrf> routing static
```

### ipv4-route

List of IPv4 static routes.

```
vrouter running config# vrf <vrf> routing static ipv4-route <ipv4-route>
```

|              |  |
|--------------|--|
| <ipv4-route> | An IPv4 prefix: address and CIDR mask. |
|--------------|--|

### next-hop

Route next-hops.

```
vrouter running config# vrf <vrf> routing static ipv4-route <ipv4-route>
vrouter running ipv4-route <ipv4-route># next-hop <next-hop> dhcp-port DHCP-PORT \
... distance <uint8> label <string> nexthop-vrf NEXTHOP-VRF tag <uint32> track TRACK
```

| <next-hop> values  | Description   |
|--------------------|---|
| <A.B.C.D>          | An IPv4 address.  |
| <ifname>           | An interface name.  |
| <A.B.C.D>%<ifname> | An IPv4 address followed by an interface name.                            |
| blackhole          | Silently discard packets when matched.                                    |
| reject             | Emit an ICMP unreachable when matched.                                    |
| dhcp-gateway       | Use the gateway acquired by DHCP on the port specified in dhcp-port leaf. |

### dhcp-port

The dhcp port, for dhcp-gateway type next-hops.

```
dhcp-port DHCP-PORT
```

|           |                    |
|-----------|--------------------|
| DHCP-PORT | An interface name. |
|-----------|--------------------|

### distance

Distance value for this route.

```
distance <uint8>
```

### label

Specify label(s) for this route. One or more labels in the range (16-1048575) separated by '/'.

```
label <string>
```

### nexthop-vrf

Nexthop vrf.

```
nexthop-vrf NEXTHOP-VRF
```

| NEXTHOP-VRF values | Description   |
|--------------------|---------------|
| main               | The main vrf. |
| <string>           | The vrf name. |

### tag

Route tag.

```
tag <uint32>
```

### track

A tracker name. If the tracked address is reachable, the next-hop is considered as valid, else it is disabled.

```
track TRACK
```

| TRACK values   | Description      |
|----------------|------------------|
| <tracker-name> | An tracker name. |
| <identityref>  | No description.  |

### ipv6-route

List of IPv6 static routes.

```
vrouter running config# vrf <vrf> routing static ipv6-route <ipv6-route>
```

|              |  |
|--------------|--|
| <ipv6-route> | An IPv6 prefix: address and CIDR mask. |
|--------------|--|

### next-hop

Route next-hops.

```
vrouter running config# vrf <vrf> routing static ipv6-route <ipv6-route>
vrouter running ipv6-route <ipv6-route># next-hop <next-hop> distance <uint8> \
... label <string> nexthop-vrf NEXTHOP-VRF tag <uint32> track TRACK
```

| <next-hop> values   | Description                                    |
|---------------------|--|
| <X:X::X:X>          | An IPv6 address.                               |
| <ifname>            | An interface name.                             |
| <X:X::X:X>%<ifname> | An IPv6 address followed by an interface name. |
| blackhole           | Silently discard packets when matched.         |
| reject              | Emit an ICMP unreachable when matched.         |

**distance**

Distance value for this route.

```
distance <uint8>
```

**label**

Specify label(s) for this route. One or more labels in the range (16-1048575) separated by '/'.

```
label <string>
```

**nexthop-vrf**

Nexthop vrf.

```
nexthop-vrf NEXTHOP-VRF
```

| NEXTHOP-VRF values | Description   |
|--------------------|---------------|
| main               | The main vrf. |
| <string>           | The vrf name. |

**tag**

Route tag.

```
tag <uint32>
```

**track**

A tracker name. If the tracked address is reachable, the next-hop is considered as valid, else it is disabled.

```
track TRACK
```

| TRACK values   | Description      |
|----------------|------------------|
| <tracker-name> | An tracker name. |
| <identityref>  | No description.  |

### table

List of routing tables.

```
vrouter running config# vrf <vrf> routing static table <uint32>
```

|          |               |
|----------|---------------|
| <uint32> | Table number. |
|----------|---------------|

### ipv4-route

List of IPv4 static routes.

```
vrouter running config# vrf <vrf> routing static table <uint32> ipv4-route <ipv4-route>
```

|              |  |
|--------------|--|
| <ipv4-route> | An IPv4 prefix: address and CIDR mask. |
|--------------|--|

### next-hop

Route next-hops.

```
vrouter running config# vrf <vrf> routing static table <uint32> ipv4-route <ipv4-route>
vrouter running ipv4-route <ipv4-route># next-hop <next-hop> dhcp-port DHCP-PORT \
... distance <uint8> label <string> nexthop-vrf NEXTHOP-VRF tag <uint32> track TRACK
```

| <next-hop> values  | Description   |
|--------------------|---|
| <A.B.C.D>          | An IPv4 address.  |
| <ifname>           | An interface name.  |
| <A.B.C.D>%<ifname> | An IPv4 address followed by an interface name.                            |
| blackhole          | Silently discard packets when matched.                                    |
| reject             | Emit an ICMP unreachable when matched.                                    |
| dhcp-gateway       | Use the gateway acquired by DHCP on the port specified in dhcp-port leaf. |

### dhcp-port

The dhcp port, for dhcp-gateway type next-hops.

```
dhcp-port DHCP-PORT
```

|           |                    |
|-----------|--------------------|
| DHCP-PORT | An interface name. |
|-----------|--------------------|

**distance**

Distance value for this route.

```
distance <uint8>
```

**label**

Specify label(s) for this route. One or more labels in the range (16-1048575) separated by '/'.

```
label <string>
```

**nexthop-vrf**

Nexthop vrf.

```
nexthop-vrf NEXTHOP-VRF
```

| NEXTHOP-VRF values | Description   |
|--------------------|---------------|
| main               | The main vrf. |
| <string>           | The vrf name. |

**tag**

Route tag.

```
tag <uint32>
```

**track**

A tracker name. If the tracked address is reachable, the next-hop is considered as valid, else it is disabled.

```
track TRACK
```

| TRACK values   | Description      |
|----------------|------------------|
| <tracker-name> | An tracker name. |
| <identityref>  | No description.  |

### ipv6-route

List of IPv6 static routes.

```
vrouter running config# vrf <vrf> routing static table <uint32> ipv6-route <ipv6-route>
```

|              |  |
|--------------|--|
| <ipv6-route> | An IPv6 prefix: address and CIDR mask. |
|--------------|--|

### next-hop

Route next-hops.

```
vrouter running config# vrf <vrf> routing static table <uint32> ipv6-route <ipv6-route>
vrouter running ipv6-route <ipv6-route># next-hop <next-hop> distance <uint8> \
... label <string> nexthop-vrf NEXTHOP-VRF tag <uint32> track TRACK
```

| <next-hop> values   | Description                                    |
|---------------------|--|
| <X:X::X:X>          | An IPv6 address.                               |
| <ifname>            | An interface name.                             |
| <X:X::X:X>%<ifname> | An IPv6 address followed by an interface name. |
| blackhole           | Silently discard packets when matched.         |
| reject              | Emit an ICMP unreachable when matched.         |

### distance

Distance value for this route.

```
distance <uint8>
```

### label

Specify label(s) for this route. One or more labels in the range (16-1048575) separated by '/'.

```
label <string>
```

## nexthop-vrf

Nexthop vrf.

```
nexthop-vrf NEXTHOP-VRF
```

| NEXTHOP-VRF values | Description   |
|--------------------|---------------|
| main               | The main vrf. |
| <string>           | The vrf name. |

## tag

Route tag.

```
tag <uint32>
```

## track

A tracker name. If the tracked address is reachable, the next-hop is considered as valid, else it is disabled.

```
track TRACK
```

| TRACK values   | Description      |
|----------------|------------------|
| <tracker-name> | An tracker name. |
| <identityref>  | No description.  |

## interface

### ip nhrp

---

**Note:** requires a Turbo Router Network License.

---

Interface NHRP configuration.

```
vrouter running config# vrf <vrf> routing interface <interface> ip nhrp
```

## enabled

Enable or disable NHRP IPv4 on this interface.

```
vrouter running config# vrf <vrf> routing interface <interface> ip nhrp
vrouter running nhrp# enabled true|false
```

### Default value

true

## registration-no-unique

Registration configuration with unique flag not set.

```
vrouter running config# vrf <vrf> routing interface <interface> ip nhrp
vrouter running nhrp# registration-no-unique true|false
```

### Default value

false

## shortcut

Allow shortcut establishment.

```
vrouter running config# vrf <vrf> routing interface <interface> ip nhrp
vrouter running nhrp# shortcut true|false
```

### Default value

false

## redirect

Send redirect notifications.

```
vrouter running config# vrf <vrf> routing interface <interface> ip nhrp
vrouter running nhrp# redirect true|false
```

### Default value

false

## shortcut-keep-sa

Do not flush IPsec SAs after shortcut expires.

```
vrouter running config# vrf <vrf> routing interface <interface> ip nhrp
vrouter running nhrp# shortcut-keep-sa true|false
```

### Default value

false

## network-id

Specify network-id to specify interface group.

```
vrouter running config# vrf <vrf> routing interface <interface> ip nhrp
vrouter running nhrp# network-id <uint32>
```

## holdtime

Time in seconds that NBMA addresses are advertised valid.

```
vrouter running config# vrf <vrf> routing interface <interface> ip nhrp
vrouter running nhrp# holdtime <uint16>
```

### Default value

7200

## ip-nhrp-mtu

MTU configuration.

```
vrouter running config# vrf <vrf> routing interface <interface> ip nhrp
vrouter running nhrp# ip-nhrp-mtu IP-NHRP-MTU
```

| IP-NHRP-MTU values | Description  |
|--------------------|--|
| <uint32>           | No description.                                    |
| opennhrp           | Advertise bound interface MTU similar to OpenNHRP. |
| default            | MTU is not configured.                             |

### Default value

default

## nhrp-map

Nexthop Server mapping configuration.

```
vrouter running config# vrf <vrf> routing interface <interface> ip nhrp nhrp-map <nhrp-
↵map>
```

|            |                  |
|------------|------------------|
| <nhrp-map> | An IPv4 address. |
|------------|------------------|

## nbma

IPv4 NBMA address.

```
vrouter running config# vrf <vrf> routing interface <interface> ip nhrp nhrp-map <nhrp-
↵map>
vrouter running nhrp-map <nhrp-map># nbma NBMA
```

| NBMA values | Description                      |
|-------------|----------------------------------|
| <A.B.C.D>   | An IPv4 address.                 |
| local       | Handle protocol address locally. |

## nhrp-nhs

Nexthop Server mapping configuration.

```
vrouter running config# vrf <vrf> routing interface <interface> ip nhrp nhrp-nhs <nhrp-
↵nhs>
```

| <nhrp-nhs> values | Description                              |
|-------------------|--|
| <A.B.C.D>         | An IPv4 address.                         |
| dynamic           | Automatic detection of protocol address. |

## nbma (mandatory)

NBMA address configuration.

```
vrouter running config# vrf <vrf> routing interface <interface> ip nhrp nhrp-nhs <nhrp-
↵nhs>
vrouter running nhrp-nhs <nhrp-nhs># nbma NBMA
```

| NBMA values | Description      |
|-------------|------------------|
| <A.B.C.D>   | An IPv4 address. |
| <string>    | No description.  |

## ipv6 nhrp

---

**Note:** requires a Turbo Router Network License.

---

Interface NHRP configuration.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 nhrp
```

### enabled

Enable or disable NHRP IPv6 on this interface.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 nhrp
vrouter running nhrp# enabled true|false
```

### Default value

true

### registration-no-unique

Registration configuration with unique flag not set.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 nhrp
vrouter running nhrp# registration-no-unique true|false
```

### Default value

false

## shortcut

Allow shortcut establishment.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 nhrp  
vrouter running nhrp# shortcut true|false
```

### Default value

false

## redirect

Send redirect notifications.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 nhrp  
vrouter running nhrp# redirect true|false
```

### Default value

false

## shortcut-keep-sa

Do not flush IPsec SAs after shortcut expires.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 nhrp  
vrouter running nhrp# shortcut-keep-sa true|false
```

### Default value

false

## network-id

Specify network-id to specify interface group.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 nhrp  
vrouter running nhrp# network-id <uint32>
```

## holdtime

Time in seconds that NBMA addresses are advertised valid.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 nhrp
vrouter running nhrp# holdtime <uint16>
```

### Default value

7200

## nhrp-map

Nexthop Server mapping configuration.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 nhrp nhrp-map
↪<nhrp-map>
```

|            |                  |
|------------|------------------|
| <nhrp-map> | An IPv6 address. |
|------------|------------------|

## nbma (mandatory)

IPv4 NBMA address.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 nhrp nhrp-map
↪<nhrp-map>
vrouter running nhrp-map <nhrp-map># nbma NBMA
```

| NBMA values | Description                      |
|-------------|----------------------------------|
| <A.B.C.D>   | An IPv4 address.                 |
| local       | Handle protocol address locally. |

## nhrp-nhs

Nexthop Server mapping configuration.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 nhrp nhrp-nhs
↪<nhrp-nhs>
```

| <nhrp-nhs> values | Description                              |
|-------------------|--|
| <X:X::X:X>        | An IPv6 address.                         |
| dynamic           | Automatic detection of protocol address. |

## nbma (mandatory)

NBMA address configuration.

```

vrouter running config# vrf <vrf> routing interface <interface> ipv6 nhrp nhrp-nhs
↪<nhrp-nhs>
vrouter running nhrp-nhs <nhrp-nhs># nbma NBMA

```

| NBMA values | Description      |
|-------------|------------------|
| <A.B.C.D>   | An IPv4 address. |
| <string>    | No description.  |

## ip ospf

**Note:** requires a Turbo Router Network License.

OSPF configuration.

```

vrouter running config# vrf <vrf> routing interface <interface> ip ospf

```

## area

OSPF area ID.

```

vrouter running config# vrf <vrf> routing interface <interface> ip ospf
vrouter running ospf# area AREA

```

| AREA values | Description      |
|-------------|------------------|
| <uint32>    | No description.  |
| <A.B.C.D>   | An IPv4 address. |

## authentication

Enable authentication on this interface.

```

vrouter running config# vrf <vrf> routing interface <interface> ip ospf
vrouter running ospf# authentication AUTHENTICATION

```

| AUTHENTICATION values | Description                        |
|-----------------------|------------------------------------|
| simple                | Use simple authentication.         |
| message-digest        | Use message-digest authentication. |
| null                  | Use null authentication.           |

### authentication-key

Authentication key.

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf
vrouter running ospf# authentication-key <string>
```

### cost

Interface cost.

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf
vrouter running ospf# cost <uint16>
```

### hello-interval

Time between HELLO packets (seconds).

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf
vrouter running ospf# hello-interval <uint16>
```

### mtu-ignore

If true, disable MTU mismatch detection on this interface.

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf
vrouter running ospf# mtu-ignore true|false
```

## priority

Router priority.

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf
vrouter running ospf# priority <uint8>
```

## retransmit-interval

Time between retransmitting lost link state advertisements (seconds).

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf
vrouter running ospf# retransmit-interval <uint16>
```

## transmit-delay

Link state transmit delay (seconds).

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf
vrouter running ospf# transmit-delay <uint16>
```

## network

Network type.

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf
vrouter running ospf# network NETWORK
```

| NETWORK values      | Description                                  |
|---------------------|--|
| broadcast           | Specify OSPF broadcast multi-access network. |
| non-broadcast       | Specify OSPF NBMA network.                   |
| point-to-multipoint | Specify OSPF point-to-multipoint network.    |
| point-to-point      | Specify OSPF point-to-point network.         |

### Default value

broadcast

## message-digest-key

Message digest authentication password (key).

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf  
vrouter running ospf# message-digest-key <uint8> md5 <string>
```

|         |         |
|---------|---------|
| <uint8> | Key ID. |
|---------|---------|

## md5 (mandatory)

The OSPF password (key).

```
md5 <string>
```

## dead-interval

Interval time after which a neighbor is declared down.

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf  
vrouter running ospf# dead-interval seconds <uint16> \  
... minimal hello-multiplier <uint8>
```

## seconds

Seconds.

```
seconds <uint16>
```

## minimal

Minimal 1s dead-interval with fast sub-second hellos.

```
minimal hello-multiplier <uint8>
```

### hello-multiplier (mandatory)

Number of Hellos to send each second.

```
hello-multiplier <uint8>
```

### address

Specific configuration per IP address.

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf address  
↳<address>
```

|           |                  |
|-----------|------------------|
| <address> | An IPv4 address. |
|-----------|------------------|

### area

OSPF area ID.

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf address  
↳<address>  
vrouter running address <address># area AREA
```

| AREA values | Description      |
|-------------|------------------|
| <uint32>    | No description.  |
| <A.B.C.D>   | An IPv4 address. |

### authentication

Enable authentication on this interface.

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf address  
↳<address>  
vrouter running address <address># authentication AUTHENTICATION
```

| AUTHENTICATION values | Description                        |
|-----------------------|------------------------------------|
| simple                | Use simple authentication.         |
| message-digest        | Use message-digest authentication. |
| null                  | Use null authentication.           |

## authentication-key

Authentication key.

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf address  
↳<address>  
vrouter running address <address># authentication-key <string>
```

## cost

Interface cost.

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf address  
↳<address>  
vrouter running address <address># cost <uint16>
```

## hello-interval

Time between HELLO packets (seconds).

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf address  
↳<address>  
vrouter running address <address># hello-interval <uint16>
```

## mtu-ignore

If true, disable MTU mismatch detection on this interface.

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf address  
↳<address>  
vrouter running address <address># mtu-ignore true|false
```

## priority

Router priority.

```
vrouter running config# vrf <vrf> routing interface <interface> ip ospf address  
↳<address>  
vrouter running address <address># priority <uint8>
```

## retransmit-interval

Time between retransmitting lost link state advertisements (seconds).

```

vrouter running config# vrf <vrf> routing interface <interface> ip ospf address
↳<address>
vrouter running address <address># retransmit-interval <uint16>

```

## transmit-delay

Link state transmit delay (seconds).

```

vrouter running config# vrf <vrf> routing interface <interface> ip ospf address
↳<address>
vrouter running address <address># transmit-delay <uint16>

```

## message-digest-key

Message digest authentication password (key).

```

vrouter running config# vrf <vrf> routing interface <interface> ip ospf address
↳<address>
vrouter running address <address># message-digest-key <uint8> md5 <string>

```

|         |         |
|---------|---------|
| <uint8> | Key ID. |
|---------|---------|

## md5 (mandatory)

The OSPF password (key).

```
md5 <string>
```

## dead-interval

Interval time after which a neighbor is declared down.

```

vrouter running config# vrf <vrf> routing interface <interface> ip ospf address
↳<address>
vrouter running address <address># dead-interval seconds <uint16> \
... minimal hello-multiplier <uint8>

```

## seconds

Seconds.

```
seconds <uint16>
```

## minimal

Minimal 1s dead-interval with fast sub-second hellos.

```
minimal hello-multiplier <uint8>
```

## hello-multiplier (mandatory)

Number of Hellos to send each second.

```
hello-multiplier <uint8>
```

## ipv6 ospf6

---

**Note:** requires a Turbo Router Network License.

---

Interface OSPFv3 configuration.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ospf6
```

## cost

Outgoing metric of this interface.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ospf6  
vrouter running ospf6# cost <uint16>
```

## dead-interval

Interval time (in seconds) after which a neighbor is declared down.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ospf6  
vrouter running ospf6# dead-interval <uint16>
```

## hello-interval

Time between HELLO packets (seconds).

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ospf6  
vrouter running ospf6# hello-interval <uint16>
```

## ifmtu

OSPFv3 Interface MTU.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ospf6  
vrouter running ospf6# ifmtu <uint16>
```

## instance-id

Instance ID for this interface.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ospf6  
vrouter running ospf6# instance-id <uint8>
```

## mtu-ignore

Disable MTU mismatch detection on this interface.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ospf6  
vrouter running ospf6# mtu-ignore true|false
```

## network

Network type.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ospf6
vrouter running ospf6# network NETWORK
```

| NETWORK values | Description                           |
|----------------|---------------------------------------|
| broadcast      | Specify OSPF6 broadcast network.      |
| point-to-point | Specify OSPF6 point-to-point network. |

### Default value

broadcast

## passive

Passive interface; no adjacency will be formed on this interface.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ospf6
vrouter running ospf6# passive true|false
```

## priority

Router priority.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ospf6
vrouter running ospf6# priority <uint8>
```

## retransmit-interval

Time between retransmitting lost link state advertisements (in seconds).

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ospf6
vrouter running ospf6# retransmit-interval <uint16>
```

## transmit-delay

Link state transmit delay (in seconds).

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ospf6
vrouter running ospf6# transmit-delay <uint16>
```

## advertise

Advertising options.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ospf6
vrouter running ospf6# advertise prefix-list <string>
```

## prefix-list (mandatory)

Filter prefix using prefix-list.

```
prefix-list <string>
```

## ip rip

---

**Note:** requires a Turbo Router Network License.

---

RIP configuration.

```
vrouter running config# vrf <vrf> routing interface <interface> ip rip
```

## v2-broadcast

Send IP broadcast v2 update.

```
vrouter running config# vrf <vrf> routing interface <interface> ip rip
vrouter running rip# v2-broadcast true|false
```

## Default value

false

## split-horizon

Controls RIP split-horizon processing on the specified interface.

```
vrouters running config# vrf <vrf> routing interface <interface> ip rip
vrouters running rip# split-horizon SPLIT-HORIZON
```

| SPLIT-HORIZON values | Description   |
|----------------------|---|
| disabled             | Disables split-horizon processing.                    |
| simple               | Enables simple split-horizon processing.              |
| poisoned-reverse     | Enables split-horizon processing with poison reverse. |

### Default value

simple

## version

Set advertisement reception/transmission version.

```
vrouters running config# vrf <vrf> routing interface <interface> ip rip version
```

## receive

Advertisement reception - Version control.

```
vrouters running config# vrf <vrf> routing interface <interface> ip rip version
vrouters running version# receive RECEIVE
```

| RECEIVE values | Description                                      |
|----------------|--|
| inherit        | Inherit configuration from the routing instance. |
| 1              | Accept RIPv1 updates only.                       |
| 2              | Accept RIPv2 updates only.                       |
| both           | Accept both RIPv1 and RIPv2 updates.             |
| none           | Do not accept neither RIPv1 nor RIPv2 updates.   |

### Default value

inherit

## send

Advertisement transmission - Version control.

```
vrouter running config# vrf <vrf> routing interface <interface> ip rip version
vrouter running version# send SEND
```

| SEND values | Description                                      |
|-------------|--|
| inherit     | Inherit configuration from the routing instance. |
| 1           | Send RIPv1 updates only.                         |
| 2           | Send RIPv2 updates only.                         |
| both        | Send both RIPv1 and RIPv2 updates.               |

### Default value

inherit

## authentication

Specify the authentication scheme for the RIP interface.

```
vrouter running config# vrf <vrf> routing interface <interface> ip rip authentication
```

## mode

Specify the authentication mode.

```
vrouter running config# vrf <vrf> routing interface <interface> ip rip authentication
vrouter running authentication# mode MODE
```

| MODE values | Description                |
|-------------|----------------------------|
| none        | No authentication.         |
| plain-text  | Plain-text authentication. |
| md5         | MD5 authentication.        |

### Default value

none

## md5-auth-length

MD5 authentication data length.

```
vrouter running config# vrf <vrf> routing interface <interface> ip rip authentication
vrouter running authentication# md5-auth-length MD5-AUTH-LENGTH
```

| MD5-AUTH-LENGTH values | Description          |
|------------------------|----------------------|
| rfc                    | RFC compatible.      |
| old-ripd               | Old ripd compatible. |

### Default value

old-ripd

## password

Authentication string.

```
vrouter running config# vrf <vrf> routing interface <interface> ip rip authentication
vrouter running authentication# password <string>
```

## key-chain

Key-chain name.

```
vrouter running config# vrf <vrf> routing interface <interface> ip rip authentication
vrouter running authentication# key-chain <string>
```

## ipv6 ripng

---

**Note:** requires a Turbo Router Network License.

---

RIPng configuration.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ripng
```

## split-horizon

Controls RIP split-horizon processing on the specified interface.

```
vrouter running config# vrf <vrf> routing interface <interface> ipv6 ripng
vrouter running ripng# split-horizon SPLIT-HORIZON
```

| SPLIT-HORIZON values | Description   |
|----------------------|---|
| disabled             | Disables split-horizon processing.                    |
| simple               | Enables simple split-horizon processing.              |
| poisoned-reverse     | Enables split-horizon processing with poison reverse. |

### Default value

simple

## evpn (state only)

---

**Note:** requires a Turbo Router Network License.

---

Operational EVPN state.

## vni (state only)

Operational EVPN state for a specific VNI.

## type (state only)

VNI type (L2/L3).

```
vrouter> show state vrf <vrf> routing evpn vni <vni> type
```

## vxlan (state only)

VXLAN name.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> vxlan
```

### num-mac (state only)

Number of mac entries.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> num-mac
```

### num-arp-nd (state only)

Number of ARP and neighbor discovery entries.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> num-arp-nd
```

### local-vtep-ip (state only)

Local virtual termination end point IP address.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> local-vtep-ip
```

### advertise-gateway-mac-ip (state only)

Advertise gateway MAC IP.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> advertise-gateway-mac-ip
```

### state (state only)

VNI state.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> state
```

### router-mac (state only)

Router MAC address.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> router-mac
```

### **l2-vni (state only)**

List of L2 VNIs linked.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> l2-vni
```

### **arp-neighbor-discovery (state only)**

List of ARP/neighbor discovery entries.

### **state (state only)**

ARP/Neighbor discovery entry state.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> arp-neighbor-discovery <arp-  
↵neighbor-discovery> state
```

### **default-gateway (state only)**

Default gateway.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> arp-neighbor-discovery <arp-  
↵neighbor-discovery> default-gateway
```

### **mac (state only)**

Entry mac address.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> arp-neighbor-discovery <arp-  
↵neighbor-discovery> mac
```

### **type (state only)**

ARP/Neighbor discovery entry type.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> arp-neighbor-discovery <arp-  
↵neighbor-discovery> type
```

### remote-vtep-ip (state only)

Remote virtual termination end point IP address.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> arp-neighbor-discovery <arp-neighbor-discovery> remote-vtep-ip
```

### mac (state only)

List of MAC entries.

### vlan-id (state only)

VLAN identifier.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> mac <mac> vlan-id
```

### type (state only)

ARP/Neighbor discovery entry type.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> mac <mac> type
```

### remote-vtep-ip (state only)

Remote virtual termination end point IP address.

```
vrouter> show state vrf <vrf> routing evpn vni <vni> mac <mac> remote-vtep-ip
```

## bgp

---

**Note:** requires a Turbo Router Network License.

---

BGP router configuration.

```
vrouter running config# vrf <vrf> routing bgp
```

## enabled

Enable or disable BGP router.

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# enabled true|false
```

### Default value

true

## as (mandatory)

BGP AS number.

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# as AS
```

|    |   |
|----|---|
| AS | A numeric identifier for an autonomous system (AS). An AS is a single domain, under common administrative control, which forms a unit of routing policy. Autonomous systems can be assigned a 2-byte identifier, or a 4-byte identifier which may have public or private scope. Private ASNs are assigned from dedicated ranges. Public ASNs are assigned from ranges allocated by IANA to the regional internet registries (RIRs). |
|----|---|

## always-compare-med

If true, allow comparing MED from different neighbors.

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# always-compare-med true|false
```

### Default value

false

## cluster-id

Configure Route-Reflector Cluster-id.

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# cluster-id CLUSTER-ID
```

| CLUSTER-ID values | Description      |
|-------------------|------------------|
| <A.B.C.D>         | An IPv4 address. |
| <uint32>          | No description.  |

### coalesce-time

Subgroup coalesce timer (in ms).

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# coalesce-time <uint32>
```

### deterministic-med

If true, Pick the best-MED path among paths advertised from the neighboring AS.

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# deterministic-med true|false
```

#### Default value

false

### ebgp-connected-route-check

Enable or disable checking if nexthop is connected on ebgp sessions.

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# ebgp-connected-route-check true|false
```

#### Default value

true

### fast-external-failover

If true, immediately reset session if a link to a directly connected external peer goes down.

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# fast-external-failover true|false
```

#### Default value

true

### graceful-shutdown

Enable or disable graceful shutdown. When enabled, EBGP route attributes are sent with the GRACEFUL\_SHUTDOWN (see RFC8326) community and preference set to 0.

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# graceful-shutdown true|false
```

#### Default value

false

### log-neighbor-changes

If true, log neighbor up/down and reset reason.

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# log-neighbor-changes true|false
```

#### Default value

false

### network-import-check

If true, check BGP network route exists in IGP.

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# network-import-check true|false
```

#### Default value

false

### route-reflector-allow-outbound-policy

If true, allow modifications made by out route-map on IBGP neighbors.

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# route-reflector-allow-outbound-policy true|false
```

#### Default value

false

### reject-as-sets

Some BGP routers may perform route aggregation, and because of that, those routers may use AS\_SET and AS\_CONFED\_SETS attributes that contain an unordered list of ASes that contributing prefixes in the aggregate have traversed. Using those attributes may cause operational issues, because they blur the semantic of origin AS.

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# reject-as-sets true|false
```

#### Default value

false

### router-id

Router id of the router.

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# router-id ROUTER-ID
```

|           |                  |
|-----------|------------------|
| ROUTER-ID | An IPv4 address. |
|-----------|------------------|

### ebgp-requires-policy

If true, require in and out policy for eBGP peers (RFC8212).

```
vrouter running config# vrf <vrf> routing bgp
vrouter running bgp# ebgp-requires-policy true|false
```

#### Default value

false

### neighbor-total-count (state only)

Total number of neighbors.

```
vrouter> show state vrf <vrf> routing bgp neighbor-total-count
```

## bestpath

Change the default bestpath selection.

```
vrouter running config# vrf <vrf> routing bgp bestpath
```

## compare-routerid

If true, compare router-id for identical EBGp paths.

```
vrouter running config# vrf <vrf> routing bgp bestpath
vrouter running bestpath# compare-routerid true|false
```

### Default value

false

## med

MED attribute.

```
vrouter running config# vrf <vrf> routing bgp bestpath
vrouter running bestpath# med MED
```

| MED values       | Description                                   |
|------------------|---|
| confederation    | Compare MED among confederation paths.        |
| missing-as-worst | Treat missing MED as the least preferred one. |

## as-path

AS-path attribute.

```
vrouter running config# vrf <vrf> routing bgp bestpath as-path
```

## confederation

If true, compare path lengths including confederation sets and sequences in selecting a route.

```
vrouter running config# vrf <vrf> routing bgp bestpath as-path
vrouter running as-path# confederation true|false
```

### Default value

false

## ignore

If true, ignore as-path length in selecting a route.

```
vrouter running config# vrf <vrf> routing bgp bestpath as-path
vrouter running as-path# ignore true|false
```

### Default value

false

## multipath-relax

Allow load sharing across routes that have different AS paths (but same length).

```
vrouter running config# vrf <vrf> routing bgp bestpath as-path
vrouter running as-path# multipath-relax MULTIPATH-RELAX
```

| MULTIPATH-RELAX values | Description                |
|------------------------|----------------------------|
| as-set                 | Generate an AS_SET.        |
| no-as-set              | Do not generate an AS_SET. |

## client-to-client

BGP client to client route reflection.

```
vrouter running config# vrf <vrf> routing bgp client-to-client
```

## reflection

Enable or disable BGP client to client route reflection.

```
vrouter running config# vrf <vrf> routing bgp client-to-client  
vrouter running client-to-client# reflection true|false
```

### Default value

true

## confederation

Parameters indicating whether the local system acts as part of a BGP confederation.

```
vrouter running config# vrf <vrf> routing bgp confederation
```

## identifier

Confederation AS number. Setting the AS indicates that the local-AS is part of a BGP confederation.

```
vrouter running config# vrf <vrf> routing bgp confederation  
vrouter running confederation# identifier <uint32>
```

## peers

Peer AS that are to be treated as part of the local confederation.

```
vrouter running config# vrf <vrf> routing bgp confederation  
vrouter running confederation# peers <uint32>
```

## dampening

Enable route-flap dampening.

```
vrouter running config# vrf <vrf> routing bgp dampening
```

## half-life

Half-life time for the penalty (minutes).

```
vrouter running config# vrf <vrf> routing bgp dampening  
vrouter running dampening# half-life <uint8>
```

## reuse

Value to start reusing a route.

```
vrouter running config# vrf <vrf> routing bgp dampening  
vrouter running dampening# reuse <uint16>
```

## suppress

Value to start suppressing a route.

```
vrouter running config# vrf <vrf> routing bgp dampening  
vrouter running dampening# suppress <uint16>
```

## max-suppress-time

Maximum duration to suppress a stable route (minutes).

```
vrouter running config# vrf <vrf> routing bgp dampening  
vrouter running dampening# max-suppress-time <uint8>
```

## graceful-restart

Configure graceful restart capability parameters.

```
vrouter running config# vrf <vrf> routing bgp graceful-restart
```

### preserve-fw-state

If true, sets F-bit indication that fib is preserved while doing Graceful Restart.

```
vrouter running config# vrf <vrf> routing bgp graceful-restart  
vrouter running graceful-restart# preserve-fw-state true|false
```

#### Default value

false

### restart-time

Set the time to wait to delete stale routes before a BGP open message is received.

```
vrouter running config# vrf <vrf> routing bgp graceful-restart  
vrouter running graceful-restart# restart-time <uint16>
```

#### Default value

120

### stalepath-time

Set the max time to hold onto restarting peer's stale paths.

```
vrouter running config# vrf <vrf> routing bgp graceful-restart  
vrouter running graceful-restart# stalepath-time <uint16>
```

#### Default value

360

### listen

Configure BGP listen options.

```
vrouter running config# vrf <vrf> routing bgp listen
```

## limit

Maximum number of BGP Dynamic neighbors that can be created.

```
vrouter running config# vrf <vrf> routing bgp listen
vrouter running listen# limit <uint16>
```

### Default value

100

## neighbor-range

Configure BGP dynamic neighbors listen range.

```
vrouter running config# vrf <vrf> routing bgp listen
vrouter running listen# neighbor-range <neighbor-range> neighbor-group <neighbor-group>
```

| <neighbor-range> values | Description                            |
|-------------------------|--|
| <A.B.C.D/M>             | An IPv4 prefix: address and CIDR mask. |
| <X:X::X:X/M>            | An IPv6 prefix: address and CIDR mask. |

## neighbor-group (mandatory)

Neighbor group name.

```
neighbor-group <neighbor-group>
```

## max-med

Advertise routes with max-med.

```
vrouter running config# vrf <vrf> routing bgp max-med
```

## administrative

Administratively applied, for an indefinite period.

```
vrouter running config# vrf <vrf> routing bgp max-med  
vrouter running max-med# administrative <uint32>
```

## on-startup

Effective on a startup.

```
vrouter running config# vrf <vrf> routing bgp max-med on-startup
```

## period (mandatory)

Time (seconds) period for max-med.

```
vrouter running config# vrf <vrf> routing bgp max-med on-startup  
vrouter running on-startup# period <uint32>
```

## max-med

Max MED value to be used.

```
vrouter running config# vrf <vrf> routing bgp max-med on-startup  
vrouter running on-startup# max-med <uint32>
```

## Default value

4294967295

## packet-rw-quantum

Number of packets to read/write from peer socket per I/O cycle.

```
vrouter running config# vrf <vrf> routing bgp packet-rw-quantum
```

## read

Number of packets to read from peer socket per I/O cycle.

```
vrouter running config# vrf <vrf> routing bgp packet-rw-quantum  
vrouter running packet-rw-quantum# read <uint8>
```

### Default value

10

## write

Number of packets to write from peer socket per I/O cycle.

```
vrouter running config# vrf <vrf> routing bgp packet-rw-quantum  
vrouter running packet-rw-quantum# write <uint8>
```

### Default value

64

## update-delay

Force initial delay for best-path and updates.

```
vrouter running config# vrf <vrf> routing bgp update-delay
```

## delay

Force initial delay for best-path and updates.

```
vrouter running config# vrf <vrf> routing bgp update-delay  
vrouter running update-delay# delay <uint16>
```

### Default value

0

## established-wait

Wait for peers to be established.

```
vrouter running config# vrf <vrf> routing bgp update-delay
vrouter running update-delay# established-wait <uint16>
```

## address-family

Address-families associated with the BGP configuration.

```
vrouter running config# vrf <vrf> routing bgp address-family
```

## ipv4-unicast

Configure IPv4 unicast address family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast
```

## table-map

BGP table to RIB route download filter.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast
vrouter running ipv4-unicast# table-map TABLE-MAP
```

|           |                 |
|-----------|-----------------|
| TABLE-MAP | Route map name. |
|-----------|-----------------|

## enabled

Enable or disable IPv4 unicast address family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast
vrouter running ipv4-unicast# enabled true|false
```

## Default value

true

## dampening

Enable/Disable route-flap dampening in this address family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast  
vrouter running ipv4-unicast# dampening true|false
```

## rib-count (state only)

Routing information base table count.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast rib-count
```

## neighbor-count (state only)

Number of neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast neighbor-count
```

## dynamic-neighbor-count (state only)

Number of dynamic neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast dynamic-neighbor-  
↪count
```

## network

Specify networks to announce via BGP.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast network  
↪<network>
```

|           |  |
|-----------|--|
| <network> | An IPv4 prefix: address and CIDR mask. |
|-----------|--|

## route-map

Route-map name.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast network  
↳<network>  
vrouter running network <network># route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## backdoor

If true, specify a BGP backdoor route.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast network  
↳<network>  
vrouter running network <network># backdoor true|false
```

### Default value

false

## label-index

Label index to associate with the prefix.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast network  
↳<network>  
vrouter running network <network># label-index <uint32>
```

## route (state only)

Route operational state.

### status (state only)

Route state.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> status
```

### stale (state only)

Entry declared stale because of restarting remote entity.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> stale
```

### suppressed (state only)

BGP path suppressed.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> suppressed
```

### history (state only)

This entry has been recorded for historical information.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> history
```

### damped (state only)

Flapping events occurred on that entry.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> damped
```

### bestpath (state only)

Selected route considered as the preferred one in the local routing context.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> bestpath
```

### multipath (state only)

Entry is selected and is not alone.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> multipath
```

### validity (state only)

Route validity.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> validity
```

### route-type (state only)

Internal or external route.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> route-type
```

### prefix (state only)

Route prefix.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> prefix
```

### prefix-length (state only)

Route prefix length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> prefix-length
```

### metric (state only)

Route metric.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> metric
```

### origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> origin
```

### network (state only)

Path network.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> network
```

### local-preference (state only)

Local preference.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> local-preference
```

### weight (state only)

Weight.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> weight
```

### packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> packet-length
```

### path (state only)

AS path.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> path
```

### nexthop (state only)

Route nexthop.

### address-family (state only)

Nexthop address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> nexthop <nexthop> address-family
```

### used (state only)

Nexthop used.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network <network>  
↪ route <string> nexthop <nexthop> used
```

### redistribute

Redistribute information from another routing protocol.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast
vrouter running ipv4-unicast# redistribute <redistribute> id <uint32> metric <uint32> \
... route-map ROUTE-MAP
```

| <redistribute> values | Description  |
|-----------------------|--|
| connected             | Connected routes (directly attached subnet or host). |
| kernel                | Kernel routes (not installed via the zebra RIB).     |
| ospf                  | Open Shortest Path First (OSPFv2).                   |
| rip                   | Routing Information Protocol (RIP).                  |
| static                | Statically configured routes.                        |
| table                 | Non-main Kernel Routing Table.                       |

### id

Instance or table ID.

```
id <uint32>
```

### metric

Metric for redistributed routes.

```
metric <uint32>
```

### route-map

Route-map name.

```
route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## route-target

Route target list.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast route-target
```

## redirect-import

Flow-spec redirect type route target, Import routes to this address- family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast route-target
vrouter running route-target# redirect-import REDIRECT-IMPORT
```

| REDIRECT-IMPORT values | Description                                     |
|------------------------|---|
| <string>               | Extended communities or route-target attribute. |
| <string>               | Extended communities or route-target attribute. |
| <string>               | Extended communities or route-target attribute. |

## l3vpn

Specify route-target and route-distinguisher between this address family and VPN.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast l3vpn
```

## export

For routes leaked from this address-family to VPN.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast l3vpn export
```

## vpn

Export routes from this address-family to default instance VPN RIB.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast l3vpn export
vrouter running export# vpn true|false
```

### Default value

false

### label

Label value (use auto to automatically assign a label).

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast l3vpn export
vrouter running export# label LABEL
```

| LABEL values | Description                   |
|--------------|-------------------------------|
| <uint32>     | No description.               |
| auto         | Automatically assign a label. |

### route-target

Specify route target list.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast l3vpn export
vrouter running export# route-target ROUTE-TARGET
```

| ROUTE-TARGET values | Description                                     |
|---------------------|---|
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |

### route-distinguisher

Specify route distinguisher.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast l3vpn export
vrouter running export# route-distinguisher ROUTE-DISTINGUISHER
```

| ROUTE-DISTINGUISHER values | Description   |
|----------------------------|---|
| <string>                   | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |
| <string>                   | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |

## nexthop

Specify next hop to use for VRF advertised prefixes between the current address-family and VPN.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast l3vpn export
vrouter running export# nexthop NEXTHOP
```

| NEXTHOP values | Description      |
|----------------|------------------|
| <A.B.C.D>      | An IPv4 address. |
| <X:X::X:X>     | An IPv6 address. |

## route-map

Specify route map between the current address-family and VPN.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast l3vpn export
vrouter running export# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## import

For routes leaked from VPN to this address-family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast l3vpn import
```

## vpn

Import routes to this address-family from default instance VPN RIB.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast l3vpn import
vrouter running import# vpn true|false
```

### Default value

false

## route-target

Specify route target list.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast l3vpn import
vrouter running import# route-target ROUTE-TARGET
```

| ROUTE-TARGET values | Description                                     |
|---------------------|---|
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |

## route-map

Specify route map between the current address-family and VPN.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast l3vpn import
vrouter running import# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## maximum-path

Forward packets over multiple paths.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast maximum-path
```

## ebgp

Ebgp number of paths.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast maximum-path
vrouter running maximum-path# ebgp <uint8>
```

### Default value

16

## ibgp

Ibgp number of paths.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast maximum-path  
vrouter running maximum-path# ibgp <uint8>
```

### Default value

16

## equal-cluster-length

If true, match the cluster length.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast maximum-path  
vrouter running maximum-path# equal-cluster-length true|false
```

### Default value

false

## aggregate-address

Configure BGP aggregate entries.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast  
vrouter running ipv4-unicast# aggregate-address <aggregate-address> as-set true|false \  
... summary-only true|false route-map ROUTE-MAP
```

|                     |  |
|---------------------|--|
| <aggregate-address> | An IPv4 prefix: address and CIDR mask. |
|---------------------|--|

## as-set

If true, generate AS set path information.

```
as-set true|false
```

### Default value

false

### summary-only

If true, filter more specific routes from updates.

```
summary-only true|false
```

#### Default value

false

### route-map

Apply route-map to aggregate network.

```
route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

### administrative-distance

Define administrative distance.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast
vrouter running ipv4-unicast# administrative-distance <administrative-distance> \
... distance <uint8> access-list ACCESS-LIST
```

|                           |  |
|---------------------------|--|
| <administrative-distance> | An IPv4 prefix: address and CIDR mask. |
|---------------------------|--|

### distance (mandatory)

Administrative distance.

```
distance <uint8>
```

## access-list

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

## bgp-distance

Configure BGP distance.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast bgp-distance
```

## external-routes

Distance for routes external to the AS.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast bgp-distance  
vrouter running bgp-distance# external-routes <uint8>
```

### Default value

20

## internal-routes

Distance for routes internal to the AS.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast bgp-distance  
vrouter running bgp-distance# internal-routes <uint8>
```

### Default value

200

## local-routes

Distance for local routes.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast bgp-distance  
vrouter running bgp-distance# local-routes <uint8>
```

### Default value

200

## route-flap-dampening

Enable route-flap dampening.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast route-flap-  
↪dampening
```

### enabled

Enable route flap dampening.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast route-flap-  
↪dampening  
vrouter running route-flap-dampening# enabled true|false
```

### Default value

false

## reach-decay

This value specifies the time desired for the instability metric value to reach one-half of its current value when the route is reachable. This half-life value determines the rate at which the metric value is decayed. A smaller half-life value makes a suppressed route reusable sooner than a larger value. The accumulated penalty will be reduced to half after this duration.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast route-flap-  
↪dampening  
vrouter running route-flap-dampening# reach-decay <uint8>
```

### Default value

15

## reuse-above

This is the value of the instability metric at which a suppressed route becomes unsuppressed if it is reachable but currently suppressed. The value assigned to reuse-below must be less than suppress-above.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# reuse-above <uint16>
```

### Default value

750

## suppress-above

This is the value of the instability metric at which route suppression takes place. A route is not installed in the forwarding information base (FIB), or announced even if it is reachable during the period that it is suppressed.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# suppress-above <uint16>
```

### Default value

2000

## unreach-decay

This value acts the same as reach-decay except that it specifies the rate at which the instability metric is decayed when a route is unreachable. It should have a value greater than or equal to reach-decay.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-unicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# unreach-decay <uint8>
```

### Default value

60

## ipv4-multicast

Configure IPv4 multicast address family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast
```

## table-map

BGP table to RIB route download filter.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast  
vrouter running ipv4-multicast# table-map TABLE-MAP
```

|           |                 |
|-----------|-----------------|
| TABLE-MAP | Route map name. |
|-----------|-----------------|

## enabled

Enable or disable IPv4 multicast Address Family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast  
vrouter running ipv4-multicast# enabled true|false
```

## Default value

true

## dampening

Enable/Disable route-flap dampening in this address family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast  
vrouter running ipv4-multicast# dampening true|false
```

## rib-count (state only)

Routing information base table count.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast rib-count
```

### neighbor-count (state only)

Number of neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast neighbor-count
```

### dynamic-neighbor-count (state only)

Number of dynamic neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast dynamic-  
↪neighbor-count
```

### network

Specify networks to announce via BGP.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast network  
↪<network>
```

|           |  |
|-----------|--|
| <network> | An IPv4 prefix: address and CIDR mask. |
|-----------|--|

### route-map

Route-map name.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast network  
↪<network>  
vrouter running network <network># route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

### backdoor

If true, specify a BGP backdoor route.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast network  
↪<network>  
vrouter running network <network># backdoor true|false
```

**Default value**

false

**label-index**

Label index to associate with the prefix.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast network  
↪<network>  
vrouter running network <network># label-index <uint32>
```

**route (state only)**

Route operational state.

**status (state only)**

Route state.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↪<network> route <string> status
```

**stale (state only)**

Entry declared stale because of restarting remote entity.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↪<network> route <string> stale
```

**suppressed (state only)**

BGP path suppressed.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↪<network> route <string> suppressed
```

### history (state only)

This entry has been recorded for historical information.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↪<network> route <string> history
```

### damped (state only)

Flapping events occurred on that entry.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↪<network> route <string> damped
```

### bestpath (state only)

Selected route considered as the preferred one in the local routing context.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↪<network> route <string> bestpath
```

### multipath (state only)

Entry is selected and is not alone.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↪<network> route <string> multipath
```

### validity (state only)

Route validity.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↪<network> route <string> validity
```

### route-type (state only)

Internal or external route.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↳<network> route <string> route-type
```

### prefix (state only)

Route prefix.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↳<network> route <string> prefix
```

### prefix-length (state only)

Route prefix length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↳<network> route <string> prefix-length
```

### metric (state only)

Route metric.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↳<network> route <string> metric
```

### origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↳<network> route <string> origin
```

### network (state only)

Path network.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↳<network> route <string> network
```

### local-preference (state only)

Local preference.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↳<network> route <string> local-preference
```

### weight (state only)

Weight.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↳<network> route <string> weight
```

### packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↳<network> route <string> packet-length
```

### path (state only)

AS path.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network  
↳<network> route <string> path
```

**nexthop (state only)**

Route nexthop.

**address-family (state only)**

Nexthop address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network
↳<network> route <string> nexthop <nexthop> address-family
```

**used (state only)**

Nexthop used.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network
↳<network> route <string> nexthop <nexthop> used
```

**aggregate-address**

Configure BGP aggregate entries.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast
vrouter running ipv4-multicast# aggregate-address <aggregate-address> as-set_
↳true|false \
... summary-only true|false route-map ROUTE-MAP
```

|                     |  |
|---------------------|--|
| <aggregate-address> | An IPv4 prefix: address and CIDR mask. |
|---------------------|--|

**as-set**

If true, generate AS set path information.

```
as-set true|false
```

**Default value**

false

### summary-only

If true, filter more specific routes from updates.

```
summary-only true|false
```

#### Default value

false

### route-map

Apply route-map to aggregate network.

```
route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

### administrative-distance

Define administrative distance.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast  
vrouter running ipv4-multicast# administrative-distance <administrative-distance> \  
... distance <uint8> access-list ACCESS-LIST
```

|                           |  |
|---------------------------|--|
| <administrative-distance> | An IPv4 prefix: address and CIDR mask. |
|---------------------------|--|

### distance (mandatory)

Administrative distance.

```
distance <uint8>
```

## access-list

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

## bgp-distance

Configure BGP distance.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast bgp-  
↳distance
```

## external-routes

Distance for routes external to the AS.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast bgp-  
↳distance  
vrouter running bgp-distance# external-routes <uint8>
```

### Default value

20

## internal-routes

Distance for routes internal to the AS.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast bgp-  
↳distance  
vrouter running bgp-distance# internal-routes <uint8>
```

### Default value

200

## local-routes

Distance for local routes.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast bgp-  
↳distance  
vrouter running bgp-distance# local-routes <uint8>
```

### Default value

200

## route-flap-dampening

Enable route-flap dampening.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast route-flap-  
↳dampening
```

### enabled

Enable route flap dampening.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# enabled true|false
```

### Default value

false

## reach-decay

This value specifies the time desired for the instability metric value to reach one-half of its current value when the route is reachable. This half-life value determines the rate at which the metric value is decayed. A smaller half-life value makes a suppressed route reusable sooner than a larger value. The accumulated penalty will be reduced to half after this duration.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# reach-decay <uint8>
```

### Default value

15

### reuse-above

This is the value of the instability metric at which a suppressed route becomes unsuppressed if it is reachable but currently suppressed. The value assigned to reuse-below must be less than suppress-above.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# reuse-above <uint16>
```

#### Default value

750

### suppress-above

This is the value of the instability metric at which route suppression takes place. A route is not installed in the forwarding information base (FIB), or announced even if it is reachable during the period that it is suppressed.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# suppress-above <uint16>
```

#### Default value

2000

### unreach-decay

This value acts the same as reach-decay except that it specifies the rate at which the instability metric is decayed when a route is unreachable. It should have a value greater than or equal to reach-decay.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-multicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# unreach-decay <uint8>
```

#### Default value

60

## ipv4-flowspec

Configure IPv4 Flowspec address family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-flowspec
```

## enabled

Enable or disable Flowspec Address Family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-flowspec  
vrouter running ipv4-flowspec# enabled true|false
```

## Default value

true

## local-install

Interface name.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-flowspec  
vrouter running ipv4-flowspec# local-install LOCAL-INSTALL
```

|               |                    |
|---------------|--------------------|
| LOCAL-INSTALL | An interface name. |
|---------------|--------------------|

## rib-count (state only)

Routing information base table count.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec rib-count
```

## neighbor-count (state only)

Number of neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec neighbor-count
```

### dynamic-neighbor-count (state only)

Number of dynamic neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec dynamic-  
↳neighbor-count
```

### route (state only)

Route operational state.

### to (state only)

Route destination prefix.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>↳  
↳to
```

### from (state only)

Route source prefix.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>↳  
↳from
```

### peer-id (state only)

Route state identifier.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>↳  
↳peer-id
```

### status (state only)

Route state.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>↳  
↳status
```

### stale (state only)

Entry declared stale because of restarting remote entity.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳stale
```

### suppressed (state only)

BGP path suppressed.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳suppressed
```

### history (state only)

This entry has been recorded for historical information.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳history
```

### damped (state only)

Flapping events occurred on that entry.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳damped
```

### bestpath (state only)

Selected route considered as the preferred one in the local routing context.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳bestpath
```

### multipath (state only)

Entry is selected and is not alone.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳multipath
```

### validity (state only)

Route validity.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳validity
```

### route-type (state only)

Internal or external route.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳route-type
```

### prefix (state only)

Route prefix.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳prefix
```

### prefix-length (state only)

Route prefix length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳prefix-length
```

### metric (state only)

Route metric.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳metric
```

### origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳origin
```

### network (state only)

Path network.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳network
```

### local-preference (state only)

Local preference.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳local-preference
```

### weight (state only)

Weight.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>_  
↳weight
```

### packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>↵  
↵packet-length
```

### path (state only)

AS path.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>↵  
↵path
```

### nexthop (state only)

Route nexthop.

### address-family (state only)

Nexthop address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>↵  
↵nexthop <nexthop> address-family
```

### used (state only)

Nexthop used.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route <uint32>↵  
↵nexthop <nexthop> used
```

### ipv4-labeled-unicast

Configure IPv4 labeled unicast address family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-labeled-unicast
```

## enabled

Enable or disable IPv4 labeled unicast Address Family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# enabled true|false
```

### Default value

true

## rib-count (state only)

Routing information base table count.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-labeled-unicast rib-count
```

## neighbor-count (state only)

Number of neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-labeled-unicast neighbor-
↪count
```

## dynamic-neighbor-count (state only)

Number of dynamic neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-labeled-unicast dynamic-
↪neighbor-count
```

## route-flap-dampening

Enable route-flap dampening.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-labeled-unicast ↵
↪route-flap-dampening
```

## enabled

Enable route flap dampening.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-labeled-unicast_
↳route-flap-dampening
vrouter running route-flap-dampening# enabled true|false
```

### Default value

false

## reach-decay

This value specifies the time desired for the instability metric value to reach one-half of its current value when the route is reachable. This half-life value determines the rate at which the metric value is decayed. A smaller half-life value makes a suppressed route reusable sooner than a larger value. The accumulated penalty will be reduced to half after this duration.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-labeled-unicast_
↳route-flap-dampening
vrouter running route-flap-dampening# reach-decay <uint8>
```

### Default value

15

## reuse-above

This is the value of the instability metric at which a suppressed route becomes unsuppressed if it is reachable but currently suppressed. The value assigned to reuse-below must be less than suppress-above.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-labeled-unicast_
↳route-flap-dampening
vrouter running route-flap-dampening# reuse-above <uint16>
```

### Default value

750

## suppress-above

This is the value of the instability metric at which route suppression takes place. A route is not installed in the forwarding information base (FIB), or announced even if it is reachable during the period that it is suppressed.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-labeled-unicast_
↳route-flap-dampening
vrouter running route-flap-dampening# suppress-above <uint16>
```

### Default value

2000

## unreach-decay

This value acts the same as reach-decay except that it specifies the rate at which the instability metric is decayed when a route is unreachable. It should have a value greater than or equal to reach- decay.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-labeled-unicast_
↳route-flap-dampening
vrouter running route-flap-dampening# unreach-decay <uint8>
```

### Default value

60

## ipv4-vpn

Configure IPv4 VPN address family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-vpn
```

## enabled

Enable or disable IPv4 VPN Address Family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-vpn
vrouter running ipv4-vpn# enabled true|false
```

### Default value

true

### rib-count (state only)

Routing information base table count.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn rib-count
```

### neighbor-count (state only)

Number of neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn neighbor-count
```

### dynamic-neighbor-count (state only)

Number of dynamic neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn dynamic-neighbor-
↳count
```

### route-distinguisher

Specify a network to announce via BGP.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-vpn route-
↳distinguisher <rd> <ip-prefix>
```

| <rd> val-ues | Description   |
|--------------|---|
| <string>     | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |
| <string>     | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |

|             |  |
|-------------|--|
| <ip-prefix> | An IPv4 prefix: address and CIDR mask. |
|-------------|--|

## label

VPN NLRI label.

```
vrouters running config# vrf <vrf> routing bgp address-family ipv4-vpn route-
↳distinguisher <rd> <ip-prefix>
vrouters running route-distinguisher <rd> <ip-prefix># label <uint32>
```

## route-map

Route map name.

```
vrouters running config# vrf <vrf> routing bgp address-family ipv4-vpn route-
↳distinguisher <rd> <ip-prefix>
vrouters running route-distinguisher <rd> <ip-prefix># route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## route (state only)

Route operational state.

## status (state only)

Route state.

```
vrouters> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher
↳<rd> <ip-prefix> route <string> status
```

## stale (state only)

Entry declared stale because of restarting remote entity.

```
vrouters> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher
↳<rd> <ip-prefix> route <string> stale
```

### suppressed (state only)

BGP path suppressed.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> suppressed
```

### history (state only)

This entry has been recorded for historical information.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> history
```

### damped (state only)

Flapping events occurred on that entry.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> damped
```

### bestpath (state only)

Selected route considered as the preferred one in the local routing context.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> bestpath
```

### multipath (state only)

Entry is selected and is not alone.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> multipath
```

### validity (state only)

Route validity.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> validity
```

### route-type (state only)

Internal or external route.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> route-type
```

### prefix (state only)

Route prefix.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> prefix
```

### prefix-length (state only)

Route prefix length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> prefix-length
```

### metric (state only)

Route metric.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> metric
```

### origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> origin
```

### network (state only)

Path network.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> network
```

### local-preference (state only)

Local preference.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> local-preference
```

### weight (state only)

Weight.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> weight
```

### packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> packet-length
```

### path (state only)

AS path.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> path
```

### nexthop (state only)

Route nexthop.

### address-family (state only)

Nexthop address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> nexthop <nexthop> address-family
```

### used (state only)

Nexthop used.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> nexthop <nexthop> used
```

### ipv6-unicast

Configure IPv6 unicast address family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast
```

### table-map

BGP table to RIB route download filter.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast  
vrouter running ipv6-unicast# table-map TABLE-MAP
```

|           |                 |
|-----------|-----------------|
| TABLE-MAP | Route map name. |
|-----------|-----------------|

## enabled

Enable or disable IPv6 unicast Address Family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast
vrouter running ipv6-unicast# enabled true|false
```

### Default value

true

## rib-count (state only)

Routing information base table count.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast rib-count
```

## neighbor-count (state only)

Number of neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast neighbor-count
```

## dynamic-neighbor-count (state only)

Number of dynamic neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast dynamic-neighbor-
↪count
```

## network

Specify a network to announce via BGP.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast network
↪<network>
```

|           |  |
|-----------|--|
| <network> | An IPv6 prefix: address and CIDR mask. |
|-----------|--|

## route-map

Route-map name.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast network
↳ <network>
vrouter running network <network># route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## label-index

Label index to associate with the prefix.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast network
↳ <network>
vrouter running network <network># label-index <uint32>
```

## route (state only)

Route operational state.

## status (state only)

Route state.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>
↳ route <string> status
```

## stale (state only)

Entry declared stale because of restarting remote entity.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>
↳ route <string> stale
```

### suppressed (state only)

BGP path suppressed.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> suppressed
```

### history (state only)

This entry has been recorded for historical information.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> history
```

### damped (state only)

Flapping events occurred on that entry.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> damped
```

### bestpath (state only)

Selected route considered as the preferred one in the local routing context.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> bestpath
```

### multipath (state only)

Entry is selected and is not alone.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> multipath
```

### validity (state only)

Route validity.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> validity
```

### route-type (state only)

Internal or external route.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> route-type
```

### prefix (state only)

Route prefix.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> prefix
```

### prefix-length (state only)

Route prefix length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> prefix-length
```

### metric (state only)

Route metric.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> metric
```

### origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> origin
```

### network (state only)

Path network.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> network
```

### local-preference (state only)

Local preference.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> local-preference
```

### weight (state only)

Weight.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> weight
```

### packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>  
↪ route <string> packet-length
```

**path (state only)**

AS path.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>
↳ route <string> path
```

**nexthop (state only)**

Route nexthop.

**address-family (state only)**

Nexthop address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>
↳ route <string> nexthop <nexthop> address-family
```

**used (state only)**

Nexthop used.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network <network>
↳ route <string> nexthop <nexthop> used
```

**aggregate-address**

Configure BGP aggregate entries.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast
vrouter running ipv6-unicast# aggregate-address <aggregate-address> as-set true|false \
... summary-only true|false route-map ROUTE-MAP
```

|                     |  |
|---------------------|--|
| <aggregate-address> | An IPv6 prefix: address and CIDR mask. |
|---------------------|--|

### as-set

If true, generate AS set path information.

```
as-set true|false
```

#### Default value

false

### summary-only

If true, filter more specific routes from updates.

```
summary-only true|false
```

#### Default value

false

### route-map

Apply route-map to aggregate network.

```
route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

### redistribute

Redistribute information from another routing protocol.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast
vrouter running ipv6-unicast# redistribute <redistribute> metric <uint32> \
... route-map ROUTE-MAP
```

| <redistribute> values | Description  |
|-----------------------|--|
| connected             | Connected routes (directly attached subnet or host).         |
| kernel                | Kernel routes (not installed via the zebra RIB).             |
| ospf6                 | Open Shortest Path First IPv6 (OSPFv3).                      |
| ripng                 | Routing Information Protocol next-generation (IPv6) (RIPng). |
| static                | Statically configured routes.                                |
| table                 | Non-main Kernel Routing Table.                               |

## metric

Metric for redistributed routes.

```
metric <uint32>
```

## route-map

Route-map name.

```
route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## ipv6-route-target

Route target list.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast ipv6-route-  
↪target
```

## redirect-import

Flow-spec redirect ipv6 type route target, Import routes to this address-family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast ipv6-route-  
↪target  
vrouter running ipv6-route-target# redirect-import REDIRECT-IMPORT
```

|                 |   |
|-----------------|---|
| REDIRECT-IMPORT | An IPv6 route target is a 20-octet BGP IPv6 address specific extended community serving the same function as a standard 8-octet route target only allowing for an IPv6 address as the global administrator. The format is <ipv6-address:2-octet-number>. Some valid examples are: 2001:DB8::1:6544 and 2001:DB8::5eb1:791:6b37:17958. |
|-----------------|---|

## administrative-distance

Define administrative distance.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast
vrouter running ipv6-unicast# administrative-distance <administrative-distance> \
... distance <uint8> access-list ACCESS-LIST
```

|                           |  |
|---------------------------|--|
| <administrative-distance> | An IPv6 prefix: address and CIDR mask. |
|---------------------------|--|

## distance (mandatory)

Administrative distance.

```
distance <uint8>
```

## access-list

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

## l3vpn

Specify route-target and route-distinguisher between this address family and VPN.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast l3vpn
```

## export

For routes leaked from this address-family to VPN.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast l3vpn export
```

## vpn

Export routes from this address-family to default instance VPN RIB.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast l3vpn export
vrouter running export# vpn true|false
```

### Default value

false

## label

Label value (use auto to automatically assign a label).

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast l3vpn export
vrouter running export# label LABEL
```

| LABEL values | Description                   |
|--------------|-------------------------------|
| <uint32>     | No description.               |
| auto         | Automatically assign a label. |

## route-target

Specify route target list.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast l3vpn export
vrouter running export# route-target ROUTE-TARGET
```

| ROUTE-TARGET values | Description                                     |
|---------------------|---|
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |

## route-distinguisher

Specify route distinguisher.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast l3vpn export
vrouter running export# route-distinguisher ROUTE-DISTINGUISHER
```

| ROUTE-DISTING values | Description   |
|----------------------|---|
| <string>             | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |
| <string>             | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |

### nexthop

Specify next hop to use for VRF advertised prefixes between the current address-family and VPN.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast l3vpn export
vrouter running export# nexthop NEXTHOP
```

| NEXTHOP values | Description      |
|----------------|------------------|
| <A.B.C.D>      | An IPv4 address. |
| <X:X::X:X>     | An IPv6 address. |

### route-map

Specify route map between the current address-family and VPN.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast l3vpn export
vrouter running export# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

### import

For routes leaked from VPN to this address-family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast l3vpn import
```

## vpn

Import routes to this address-family from default instance VPN RIB.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast l3vpn import
vrouter running import# vpn true|false
```

### Default value

false

## route-target

Specify route target list.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast l3vpn import
vrouter running import# route-target ROUTE-TARGET
```

| ROUTE-TARGET values | Description                                     |
|---------------------|---|
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |

## route-map

Specify route map between the current address-family and VPN.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast l3vpn import
vrouter running import# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## maximum-path

Forward packets over multiple paths.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast maximum-path
```

## ebgp

Ebgp number of paths.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast maximum-path  
vrouter running maximum-path# ebgp <uint8>
```

### Default value

16

## ibgp

Ibgp number of paths.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast maximum-path  
vrouter running maximum-path# ibgp <uint8>
```

### Default value

16

## equal-cluster-length

If true, match the cluster length.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast maximum-path  
vrouter running maximum-path# equal-cluster-length true|false
```

### Default value

false

## bgp-distance

Configure BGP distance.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast bgp-distance
```

## external-routes

Distance for routes external to the AS.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast bgp-distance  
vrouter running bgp-distance# external-routes <uint8>
```

### Default value

20

## internal-routes

Distance for routes internal to the AS.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast bgp-distance  
vrouter running bgp-distance# internal-routes <uint8>
```

### Default value

200

## local-routes

Distance for local routes.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast bgp-distance  
vrouter running bgp-distance# local-routes <uint8>
```

### Default value

200

## route-flap-dampening

Enable route-flap dampening.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast route-flap-  
↪dampening
```

## enabled

Enable route flap dampening.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# enabled true|false
```

### Default value

false

## reach-decay

This value specifies the time desired for the instability metric value to reach one-half of its current value when the route is reachable. This half-life value determines the rate at which the metric value is decayed. A smaller half-life value makes a suppressed route reusable sooner than a larger value. The accumulated penalty will be reduced to half after this duration.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# reach-decay <uint8>
```

### Default value

15

## reuse-above

This is the value of the instability metric at which a suppressed route becomes unsuppressed if it is reachable but currently suppressed. The value assigned to reuse-below must be less than suppress-above.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# reuse-above <uint16>
```

### Default value

750

## suppress-above

This is the value of the instability metric at which route suppression takes place. A route is not installed in the forwarding information base (FIB), or announced even if it is reachable during the period that it is suppressed.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# suppress-above <uint16>
```

### Default value

2000

## unreach-decay

This value acts the same as reach-decay except that it specifies the rate at which the instability metric is decayed when a route is unreachable. It should have a value greater than or equal to reach-decay.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-unicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# unreach-decay <uint8>
```

### Default value

60

## ipv6-flowspec

Configure IPv6 Flowspec address family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-flowspec
```

## enabled

Enable or disable Flowspec Address Family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-flowspec  
vrouter running ipv6-flowspec# enabled true|false
```

### Default value

true

## local-install

Interface name.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-flowspec  
vrouter running ipv6-flowspec# local-install LOCAL-INSTALL
```

|               |                    |
|---------------|--------------------|
| LOCAL-INSTALL | An interface name. |
|---------------|--------------------|

## rib-count (state only)

Routing information base table count.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec rib-count
```

## neighbor-count (state only)

Number of neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec neighbor-count
```

## dynamic-neighbor-count (state only)

Number of dynamic neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec dynamic-  
↪neighbor-count
```

## route (state only)

Route operational state.

**to (state only)**

Route destination prefix.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳to
```

**from (state only)**

Route source prefix.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳from
```

**peer-id (state only)**

Route state identifier.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳peer-id
```

**status (state only)**

Route state.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳status
```

**stale (state only)**

Entry declared stale because of restarting remote entity.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳stale
```

### suppressed (state only)

BGP path suppressed.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳suppressed
```

### history (state only)

This entry has been recorded for historical information.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳history
```

### damped (state only)

Flapping events occurred on that entry.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳damped
```

### bestpath (state only)

Selected route considered as the preferred one in the local routing context.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳bestpath
```

### multipath (state only)

Entry is selected and is not alone.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳multipath
```

### validity (state only)

Route validity.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳validity
```

### route-type (state only)

Internal or external route.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳route-type
```

### prefix (state only)

Route prefix.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳prefix
```

### prefix-length (state only)

Route prefix length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳prefix-length
```

### metric (state only)

Route metric.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳metric
```

### origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳origin
```

### network (state only)

Path network.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳network
```

### local-preference (state only)

Local preference.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳local-preference
```

### weight (state only)

Weight.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳weight
```

### packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳packet-length
```

### path (state only)

AS path.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳path
```

### nexthop (state only)

Route nexthop.

### address-family (state only)

Nexthop address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳nexthop <nexthop> address-family
```

### used (state only)

Nexthop used.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route <uint32>_  
↳nexthop <nexthop> used
```

### ipv6-multicast

Configure IPv6 multicast address family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast
```

### enabled

Enable or disable IPv6 multicast Address Family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast  
vrouter running ipv6-multicast# enabled true|false
```

### Default value

true

**rib-count (state only)**

Routing information base table count.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast rib-count
```

**neighbor-count (state only)**

Number of neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast neighbor-count
```

**dynamic-neighbor-count (state only)**

Number of dynamic neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast dynamic-  
↪neighbor-count
```

**network**

Specify a network to announce via BGP.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast network  
↪<network>
```

|           |  |
|-----------|--|
| <network> | An IPv6 prefix: address and CIDR mask. |
|-----------|--|

**route-map**

Route-map name.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast network  
↪<network>  
vrouter running network <network># route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## label-index

Label index to associate with the prefix.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast network  
↪<network>  
vrouter running network <network># label-index <uint32>
```

## route (state only)

Route operational state.

## status (state only)

Route state.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↪<network> route <string> status
```

## stale (state only)

Entry declared stale because of restarting remote entity.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↪<network> route <string> stale
```

## suppressed (state only)

BGP path suppressed.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↪<network> route <string> suppressed
```

### history (state only)

This entry has been recorded for historical information.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↪<network> route <string> history
```

### damped (state only)

Flapping events occurred on that entry.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↪<network> route <string> damped
```

### bestpath (state only)

Selected route considered as the preferred one in the local routing context.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↪<network> route <string> bestpath
```

### multipath (state only)

Entry is selected and is not alone.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↪<network> route <string> multipath
```

### validity (state only)

Route validity.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↪<network> route <string> validity
```

### route-type (state only)

Internal or external route.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↳<network> route <string> route-type
```

### prefix (state only)

Route prefix.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↳<network> route <string> prefix
```

### prefix-length (state only)

Route prefix length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↳<network> route <string> prefix-length
```

### metric (state only)

Route metric.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↳<network> route <string> metric
```

### origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↳<network> route <string> origin
```

### network (state only)

Path network.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↳<network> route <string> network
```

### local-preference (state only)

Local preference.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↳<network> route <string> local-preference
```

### weight (state only)

Weight.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↳<network> route <string> weight
```

### packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↳<network> route <string> packet-length
```

### path (state only)

AS path.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↳<network> route <string> path
```

### nexthop (state only)

Route nexthop.

### address-family (state only)

Nexthop address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↳<network> route <string> nexthop <nexthop> address-family
```

### used (state only)

Nexthop used.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network  
↳<network> route <string> nexthop <nexthop> used
```

### administrative-distance

Define administrative distance.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast  
vrouter running ipv6-multicast# administrative-distance <administrative-distance> \  
... distance <uint8> access-list ACCESS-LIST
```

|                           |  |
|---------------------------|--|
| <administrative-distance> | An IPv6 prefix: address and CIDR mask. |
|---------------------------|--|

### distance (mandatory)

Administrative distance.

```
distance <uint8>
```

## access-list

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

## bgp-distance

Configure BGP distance.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast bgp-  
↳distance
```

## external-routes

Distance for routes external to the AS.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast bgp-  
↳distance  
vrouter running bgp-distance# external-routes <uint8>
```

### Default value

20

## internal-routes

Distance for routes internal to the AS.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast bgp-  
↳distance  
vrouter running bgp-distance# internal-routes <uint8>
```

### Default value

200

## local-routes

Distance for local routes.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast bgp-
↳distance
vrouter running bgp-distance# local-routes <uint8>
```

### Default value

200

## route-flap-dampening

Enable route-flap dampening.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast route-flap-
↳dampening
```

### enabled

Enable route flap dampening.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast route-flap-
↳dampening
vrouter running route-flap-dampening# enabled true|false
```

### Default value

false

## reach-decay

This value specifies the time desired for the instability metric value to reach one-half of its current value when the route is reachable. This half-life value determines the rate at which the metric value is decayed. A smaller half-life value makes a suppressed route reusable sooner than a larger value. The accumulated penalty will be reduced to half after this duration.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast route-flap-
↳dampening
vrouter running route-flap-dampening# reach-decay <uint8>
```

### Default value

15

### reuse-above

This is the value of the instability metric at which a suppressed route becomes unsuppressed if it is reachable but currently suppressed. The value assigned to reuse-below must be less than suppress-above.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# reuse-above <uint16>
```

#### Default value

750

### suppress-above

This is the value of the instability metric at which route suppression takes place. A route is not installed in the forwarding information base (FIB), or announced even if it is reachable during the period that it is suppressed.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# suppress-above <uint16>
```

#### Default value

2000

### unreach-decay

This value acts the same as reach-decay except that it specifies the rate at which the instability metric is decayed when a route is unreachable. It should have a value greater than or equal to reach-decay.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-multicast route-flap-  
↳dampening  
vrouter running route-flap-dampening# unreach-decay <uint8>
```

#### Default value

60

## ipv6-labeled-unicast

Configure IPv6 labeled unicast address family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-labeled-unicast
```

## enabled

Enable or disable IPv6 labeled unicast Address Family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-labeled-unicast  
vrouter running ipv6-labeled-unicast# enabled true|false
```

## Default value

true

## rib-count (state only)

Routing information base table count.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-labeled-unicast rib-count
```

## neighbor-count (state only)

Number of neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-labeled-unicast neighbor-  
↪count
```

## dynamic-neighbor-count (state only)

Number of dynamic neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-labeled-unicast dynamic-  
↪neighbor-count
```

## maximum-path

Forward packets over multiple paths.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-labeled-unicast_
↳maximum-path
```

## ebgp

Ebgp number of paths.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-labeled-unicast_
↳maximum-path
vrouter running maximum-path# ebgp <uint8>
```

### Default value

16

## ibgp

Ibgp number of paths.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-labeled-unicast_
↳maximum-path
vrouter running maximum-path# ibgp <uint8>
```

### Default value

16

## equal-cluster-length

If true, match the cluster length.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-labeled-unicast_
↳maximum-path
vrouter running maximum-path# equal-cluster-length true|false
```

### Default value

false

## route-flap-dampening

Enable route-flap dampening.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-labeled-unicast_
↳route-flap-dampening
```

### enabled

Enable route flap dampening.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-labeled-unicast_
↳route-flap-dampening
vrouter running route-flap-dampening# enabled true|false
```

### Default value

false

### reach-decay

This value specifies the time desired for the instability metric value to reach one-half of its current value when the route is reachable. This half-life value determines the rate at which the metric value is decayed. A smaller half-life value makes a suppressed route reusable sooner than a larger value. The accumulated penalty will be reduced to half after this duration.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-labeled-unicast_
↳route-flap-dampening
vrouter running route-flap-dampening# reach-decay <uint8>
```

### Default value

15

### reuse-above

This is the value of the instability metric at which a suppressed route becomes unsuppressed if it is reachable but currently suppressed. The value assigned to reuse-below must be less than suppress-above.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-labeled-unicast_
↳route-flap-dampening
vrouter running route-flap-dampening# reuse-above <uint16>
```

### Default value

750

### suppress-above

This is the value of the instability metric at which route suppression takes place. A route is not installed in the forwarding information base (FIB), or announced even if it is reachable during the period that it is suppressed.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-labeled-unicast_
↳route-flap-dampening
vrouter running route-flap-dampening# suppress-above <uint16>
```

#### Default value

2000

### unreach-decay

This value acts the same as reach-decay except that it specifies the rate at which the instability metric is decayed when a route is unreachable. It should have a value greater than or equal to reach-decay.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-labeled-unicast_
↳route-flap-dampening
vrouter running route-flap-dampening# unreach-decay <uint8>
```

#### Default value

60

### ipv6-vpn

Configure IPv6 VPN address family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-vpn
```

### enabled

Enable or disable IPv6 VPN Address Family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-vpn
vrouter running ipv6-vpn# enabled true|false
```

#### Default value

true

**rib-count (state only)**

Routing information base table count.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn rib-count
```

**neighbor-count (state only)**

Number of neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn neighbor-count
```

**dynamic-neighbor-count (state only)**

Number of dynamic neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn dynamic-neighbor-  
↪count
```

**route-distinguisher**

Specify a network to announce via BGP.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-vpn route-  
↪distinguisher <rd> <ip-prefix>
```

| <rd><br>val-<br>ues | Description   |
|---------------------|---|
| <string>            | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |
| <string>            | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |

|             |  |
|-------------|--|
| <ip-prefix> | An IPv6 prefix: address and CIDR mask. |
|-------------|--|

## label

VPN NLRI label.

```
vrouters running config# vrf <vrf> routing bgp address-family ipv6-vpn route-
↳distinguisher <rd> <ip-prefix>
vrouters running route-distinguisher <rd> <ip-prefix># label <uint32>
```

## route-map

Route map name.

```
vrouters running config# vrf <vrf> routing bgp address-family ipv6-vpn route-
↳distinguisher <rd> <ip-prefix>
vrouters running route-distinguisher <rd> <ip-prefix># route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## route (state only)

Route operational state.

## status (state only)

Route state.

```
vrouters> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher
↳<rd> <ip-prefix> route <string> status
```

## stale (state only)

Entry declared stale because of restarting remote entity.

```
vrouters> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher
↳<rd> <ip-prefix> route <string> stale
```

### suppressed (state only)

BGP path suppressed.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> suppressed
```

### history (state only)

This entry has been recorded for historical information.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> history
```

### damped (state only)

Flapping events occurred on that entry.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> damped
```

### bestpath (state only)

Selected route considered as the preferred one in the local routing context.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> bestpath
```

### multipath (state only)

Entry is selected and is not alone.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> multipath
```

### validity (state only)

Route validity.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> validity
```

### route-type (state only)

Internal or external route.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> route-type
```

### prefix (state only)

Route prefix.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> prefix
```

### prefix-length (state only)

Route prefix length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> prefix-length
```

### metric (state only)

Route metric.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> metric
```

### origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> origin
```

### network (state only)

Path network.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> network
```

### local-preference (state only)

Local preference.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> local-preference
```

### weight (state only)

Weight.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> weight
```

### packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> packet-length
```

### path (state only)

AS path.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> path
```

### nexthop (state only)

Route nexthop.

### address-family (state only)

Nexthop address family.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> nexthop <nexthop> address-family
```

### used (state only)

Nexthop used.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-distinguisher  
↳<rd> <ip-prefix> route <string> nexthop <nexthop> used
```

### l2vpn-evpn

Configure L2VPN EVPN address family.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn
```

### enabled

Enable or disable L2VPN EVPN Address Family configuration.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn  
vrouter running l2vpn-evpn# enabled true|false
```

### Default value

true

## advertise

Configure advertisement.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn
vrouter running l2vpn-evpn# advertise ADVERTISE
```

| ADVERTISE values | Description                    |
|------------------|--------------------------------|
| ipv4-unicast     | Advertise IPv4 unicast routes. |
| ipv6-unicast     | Advertise IPv6 unicast routes. |

## advertise-all-vni

Advertise All local VNIs.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn
vrouter running l2vpn-evpn# advertise-all-vni true|false
```

### Default value

false

## auto-route-target

Configure auto-derivation of route-target.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn
vrouter running l2vpn-evpn# auto-route-target AUTO-ROUTE-TARGET
```

| AUTO-ROUTE-TARGET values | Description                                     |
|--------------------------|---|
| disabled                 | Do not auto-derivate route targets.             |
| rfc8365                  | Auto-derivation of route targets using RFC8365. |

### Default value

disabled

## default-originate

Originate a default route.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn
vrouter running l2vpn-evpn# default-originate DEFAULT-ORIGINATE
```

| DEFAULT-ORIGINATE values | Description                     |
|--------------------------|---------------------------------|
| ipv4                     | IPv4 address family.            |
| ipv6                     | IPv6 address family.            |
| both                     | IPv4 and IPv6 address families. |

## flooding

Specify handling for BUM packets.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn
vrouter running l2vpn-evpn# flooding FLOODING
```

| FLOODING values      | Description                                   |
|----------------------|---|
| disabled             | Do not flood any BUM packets.                 |
| head-end-replication | Flood BUM packets using head-end replication. |

### Default value

head-end-replication

## rib-count (state only)

Routing information base table count.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn rib-count
```

## neighbor-count (state only)

Number of neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn neighbor-count
```

### dynamic-neighbor-count (state only)

Number of dynamic neighbors for this address family.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn dynamic-neighbor-  
↪ count
```

### vni

Configure L2VPN for a specific VXLAN Network Identifier.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn vni <vni>
```

|       |   |
|-------|---|
| <vni> | Type definition representing VXLAN Segment ID / VXLAN Network Identifier value. |
|-------|---|

### enabled

Enable/Disable advertisement for this VNI.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn vni <vni>  
vrouter running vni <vni># enabled true|false
```

#### Default value

true

### advertise-default-gw

Advertise all default g/w mac-ip routes in EVPN.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn vni <vni>  
vrouter running vni <vni># advertise-default-gw true|false
```

#### Default value

false

### export

Configure route-target to export.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn vni <vni>
↳ export
```

### route-target

Specify the route target.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn vni <vni>
↳ export
vrouter running export# route-target ROUTE-TARGET
```

| ROUTE-TARGET values | Description                                     |
|---------------------|---|
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |

### route-distinguisher

Specify route distinguisher.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn vni <vni>
↳ export
vrouter running export# route-distinguisher ROUTE-DISTINGUISHER
```

| ROUTE-DISTINGUISHER values | Description   |
|----------------------------|---|
| <string>                   | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |
| <string>                   | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |

## import

Configure route-target to import.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn vni <vni>
↳import
```

## route-target

Specify the route target.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn vni <vni>
↳import
vrouter running import# route-target ROUTE-TARGET
```

| ROUTE-TARGET values | Description                                     |
|---------------------|---|
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |

## export

Configure route-target to export.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn export
```

## route-target

Specify the route target.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn export
vrouter running export# route-target ROUTE-TARGET
```

| ROUTE-TARGET values | Description                                     |
|---------------------|---|
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |

### route-distinguisher

Specify route distinguisher.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn export
vrouter running export# route-distinguisher ROUTE-DISTINGUISHER
```

| ROUTE-DISTINGUISHER values | Description   |
|----------------------------|---|
| <string>                   | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |
| <string>                   | Type definition for extended community attributes. In the case that common communities are utilised, they are represented as a string of the form: - <2b AS>:<4b value> per RFC4360 section 3.1 - <4b IPv4>:<2b value> per RFC4360 section 3.2. |

### import

Configure route-target to import.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn import
```

### route-target

Specify the route target.

```
vrouter running config# vrf <vrf> routing bgp address-family l2vpn-evpn import
vrouter running import# route-target ROUTE-TARGET
```

| ROUTE-TARGET values | Description                                     |
|---------------------|---|
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |
| <string>            | Extended communities or route-target attribute. |

**route-distinguisher (state only)**

Route distinguisher operational state.

**prefix (state only)**

Route distinguisher prefix list.

**prefix-length (state only)**

Route prefix length.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↪distinguisher <route-distinguisher> prefix <string> prefix-length
```

**path (state only)**

Path list.

**peer-id (state only)**

Route state identifier.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↪distinguisher <route-distinguisher> prefix <string> path <uint32> peer-id
```

**status (state only)**

Route state.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↪distinguisher <route-distinguisher> prefix <string> path <uint32> status
```

### stale (state only)

Entry declared stale because of restarting remote entity.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> stale
```

### suppressed (state only)

BGP path suppressed.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> suppressed
```

### history (state only)

This entry has been recorded for historical information.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> history
```

### damped (state only)

Flapping events occurred on that entry.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> damped
```

### bestpath (state only)

Selected route considered as the preferred one in the local routing context.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> bestpath
```

### multipath (state only)

Entry is selected and is not alone.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> multipath
```

### validity (state only)

Route validity.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> validity
```

### route-type (state only)

Internal or external route.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> route-type
```

### prefix (state only)

Route prefix.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> prefix
```

### prefix-length (state only)

Route prefix length.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> prefix-length
```

### metric (state only)

Route metric.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> metric
```

### origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> origin
```

### network (state only)

Path network.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> network
```

### local-preference (state only)

Local preference.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> local-preference
```

### weight (state only)

Weight.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> weight
```

### packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> packet-length
```

### path (state only)

AS path.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> path
```

### nexthop (state only)

Route nexthop.

### address-family (state only)

Nexthop address family.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> nexthop <nexthop>↳  
↳address-family
```

### used (state only)

Nexthop used.

```
vrouter> show state vrf <vrf> routing bgp address-family l2vpn-evpn route-  
↳distinguisher <route-distinguisher> prefix <string> path <uint32> nexthop <nexthop>↳  
↳used
```

## neighbor-group

List of BGP peer-groups configured on the local system - uniquely identified by peer-group name.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>
```

|          |   |
|----------|---|
| <string> | Reference to the name of the BGP neighbor-group used as a key in the neighbor-group list. |
|----------|---|

## remote-as

Remote AS number.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>
vrouter running neighbor-group <string># remote-as REMOTE-AS
```

| REMOTE-AS values | Description        |
|------------------|--------------------|
| <uint32>         | No description.    |
| external         | External BGP peer. |
| internal         | Internal BGP peer. |

## capability

Advertise capability to the peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>
vrouter running neighbor-group <string># capability CAPABILITY
```

| CAPABILITY values | Description  |
|-------------------|--|
| dynamic           | Advertise dynamic capability to this neighbor.     |
| extended-nexthop  | Advertise extended nexthop capability to the peer. |

## capability-negotiate

If true, perform capability negotiation.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>
vrouter running neighbor-group <string># capability-negotiate true|false
```

### Default value

true

## ebgp-multihop

Allow EBGp neighbors not on directly connected networks.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>  
vrouter running neighbor-group <string># ebgp-multihop <uint8>
```

## enforce-first-as

If true, enforce the first AS for EBGp routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>  
vrouter running neighbor-group <string># enforce-first-as true|false
```

### Default value

false

## enforce-multihop

If true, enforce EBGp neighbors perform multihop.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>  
vrouter running neighbor-group <string># enforce-multihop true|false
```

### Default value

false

## neighbor-description

Neighbor specific description: up to 80 characters describing this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>  
vrouter running neighbor-group <string># neighbor-description <string>
```

### override-capability

If true, override capability negotiation result.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>  
vrouter running neighbor-group <string># override-capability true|false
```

#### Default value

false

### passive

If true, don't send open messages to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>  
vrouter running neighbor-group <string># passive true|false
```

#### Default value

false

### password

Set a password.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>  
vrouter running neighbor-group <string># password <string>
```

### solo

If true, solo peer - part of its own update group.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>  
vrouter running neighbor-group <string># solo true|false
```

#### Default value

false

### strict-capability-match

Enable or disable strict capability negotiation match.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>
vrouter running neighbor-group <string># strict-capability-match true|false
```

#### Default value

false

### track

A tracker name defined in the tracker context, or the BGP internal BFD tracker (bfd keyword). If a tracker name is used, when the tracked address is reachable, the neighbor or neighbor group is considered as valid, else it is disabled. If the BGP internal BFD tracker is used, it works the same way, but this neighbor address is automatically tracked. The check-control-plane-failure option is available only when the BGP internal BFD tracker is used.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>
vrouter running neighbor-group <string># track TRACK
```

| TRACK values   | Description      |
|----------------|------------------|
| <tracker-name> | An tracker name. |
| <identityref>  | No description.  |

### check-control-plane-failure

Link data-plane status with BGP control-plane. This option is available only if the BGP internal BFD tracker is selected, i.e the ‘track bfd’ option is set.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>
vrouter running neighbor-group <string># check-control-plane-failure true|false
```

### ttl-security-hops

Specify the maximum number of hops to the BGP peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>
vrouter running neighbor-group <string># ttl-security-hops <uint8>
```

## update-source

Source of routing updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>
vrouter running neighbor-group <string># update-source UPDATE-SOURCE
```

| UPDATE-SOURCE values | Description        |
|----------------------|--------------------|
| <A.B.C.D>            | An IPv4 address.   |
| <X:X::X:X>           | An IPv6 address.   |
| <ifname>             | An interface name. |

## sender-as-path-loop-detection

Detect the sender side AS path loops and filter the bad routes before they are sent.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string>
vrouter running neighbor-group <string># sender-as-path-loop-detection true|false
```

### Default value

false

## remote-neighbor-group (state only)

Remote neighbor group.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> remote-neighbor-group
```

## remote-router-id (state only)

Remote router identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> remote-router-id
```

### state (state only)

BGP router status.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> state
```

### min-time-btwn-advertisement (state only)

Minimum time between advertisement runs in milliseconds.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> min-time-btwn-  
↪advertisement
```

### last-reset (state only)

Last reset.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> last-reset
```

### bgp-connection (state only)

BGP connection type.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> bgp-connection
```

### connect-retry-timer (state only)

BGP connect retry timer in seconds.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> connect-retry-timer
```

### estimated-round-trip-time (state only)

Estimated round trip time in milliseconds.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> estimated-round-trip-  
↪time
```

## local-as

Specify a local-as number.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> local-as
```

## as-number (mandatory)

AS number used as local AS.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> local-as  
vrouter running local-as# as-number <uint32>
```

## no-prepend

If true, do not prepend local-as to updates from ebgp peers.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> local-as  
vrouter running local-as# no-prepend true|false
```

### Default value

false

## replace-as

If true, do not prepend local-as to updates from ibgp peers.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> local-as  
vrouter running local-as# replace-as true|false
```

### Default value

false

## shutdown

Administratively shut down this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> shutdown
```

## message

Shutdown message.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> shutdown
vrouter running shutdown# message <string>
```

## timers

Config parameters related to timers associated with the BGP peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> timers
```

## advertisement-interval

Minimum time which must elapse between subsequent UPDATE messages relating to a common set of NLRI being transmitted to a peer. This timer is referred to as MinRouteAdvertisementIntervalTimer by RFC 4721 and serves to reduce the number of UPDATE messages transmitted when a particular set of NLRI exhibit instability. A change of this value will be taken into account for new sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> timers
vrouter running timers# advertisement-interval <uint16>
```

## connect-retry

Time interval in seconds between attempts to establish a session with the peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> timers
vrouter running timers# connect-retry <uint16>
```

## keepalive-interval

Time interval in seconds between transmission of keepalive messages to the neighbor. Typically set to 1/3 the hold-time. A change of this value will be taken into account for new sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> timers
vrouter running timers# keepalive-interval <uint16>
```

## hold-time

Time interval in seconds that a BGP session will be considered active in the absence of keepalive or other messages from the peer. The hold-time is typically set to 3x the keepalive-interval.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> timers
vrouter running timers# hold-time <uint16>
```

## address-family

Address-families associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family
```

## ipv4-unicast

IPv4 unicast address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
```

## enabled

Enable or disable IPv4 unicast Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
vrouter running ipv4-unicast# enabled true|false
```

## Default value

true

## as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
vrouter running ipv4-unicast# as-override true|false
```

## Default value

false

### maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
vrouter running ipv4-unicast# maximum-prefix-out <uint32>
```

### send-community

Send Community attribute to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
vrouter running ipv4-unicast# send-community SEND-COMMUNITY
```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

### weight

Set default weight for routes from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
vrouter running ipv4-unicast# weight <uint16>
```

## allowas-in

Accept as-path with my AS present in it.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
vrouters running ipv4-unicast# allowas-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

## attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
vrouters running ipv4-unicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

## route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
vrouters running ipv4-unicast# route-reflector-client true|false
```

### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
vrouter running ipv4-unicast# route-server-client true|false
```

#### Default value

false

### soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
vrouter running ipv4-unicast# soft-reconfiguration-inbound true|false
```

#### Default value

false

### capability-orf-prefix-list

Advertise prefixlist ORF capability to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
vrouter running ipv4-unicast# capability-orf-prefix-list CAPABILITY-ORF-PREFIX-LIST
```

| CAPABILITY-ORF-PREFIX-LIST values | Description   |
|-----------------------------------|---|
| both                              | Capability to SEND and RECEIVE the ORF to/from this neighbor. |
| send                              | Capability to SEND the ORF to this neighbor.                  |
| receive                           | Capability to RECEIVE the ORF from this neighbor.             |

## unsuppress-map

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
vrouter running ipv4-unicast# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

## update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-
↳unicast update-group-id
```

## sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-
↳unicast sub-group-id
```

## packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-
↳unicast packet-queue-length
```

## accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-
↳unicast accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-
↳unicast inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-
↳unicast outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast addpath
vrouter running addpath# tx-all-paths true|false
```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast addpath
vrouter running addpath# tx-best-path-per-AS true|false
```

#### Default value

false

### distributed-list

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
vrouter running ipv4-unicast# distribute-list <distribute-list> access-list ACCESS-LIST
```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

### access-list (mandatory)

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

### maximum-prefix

Maximum number of prefixes to accept from this peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast maximum-prefix
```

### maximum (mandatory)

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast maximum-prefix
vrouter running maximum-prefix# maximum <uint32>
```

## threshold

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast maximum-prefix
vrouter running maximum-prefix# threshold <uint8>
```

### Default value

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast maximum-prefix
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast maximum-prefix
vrouter running maximum-prefix# warning-only true|false
```

### Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast nexthop-self
vrouter running nexthop-self# force true|false

```

### Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast as-outbound-update

```

## action

Action to apply for ASNs in outbound updates.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast as-outbound-update
vrouter running as-outbound-update# action ACTION

```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast as-outbound-update
vrouter running as-outbound-update# as-type AS-TYPE

```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

**Default value**

private

**filter-list**

Establish BGP filters.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ ipv4-unicast
vrouters running ipv4-unicast# filter-list <filter-list> access-list <as-path-access-
↳ list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list <as-path-access-list>
```

**prefix-list**

Filter updates to/from this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ ipv4-unicast
vrouters running ipv4-unicast# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-
↳ NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

### prefix-list-name (mandatory)

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

### route-map

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast
vrouter running ipv4-unicast# route-map <route-map> route-map-name ROUTE-MAP-NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

### route-map-name (mandatory)

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

### default-originate

Originate default route to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast default-originate
```

## route-map

Route-map to specify criteria to originate default.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-unicast default-originate
vrouter running default-originate# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## ipv4-multicast

IPv4 multicast address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
```

## enabled

Enable or disable IPv4 multicast Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouter running ipv4-multicast# enabled true|false
```

### Default value

true

## as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouter running ipv4-multicast# as-override true|false
```

### Default value

false

## maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouters running ipv4-multicast# maximum-prefix-out <uint32>

```

## send-community

Send Community attribute to this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouters running ipv4-multicast# send-community SEND-COMMUNITY

```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

## weight

Set default weight for routes from this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouters running ipv4-multicast# weight <uint16>

```

## allowas-in

Accept as-path with my AS present in it.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouters running ipv4-multicast# allowas-in ALLOWAS-IN

```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

### attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouters running ipv4-multicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

### route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouters running ipv4-multicast# route-reflector-client true|false
```

#### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouters running ipv4-multicast# route-server-client true|false
```

#### Default value

false

## soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouter running ipv4-multicast# soft-reconfiguration-inbound true|false
```

### Default value

false

## capability-orf-prefix-list

Advertise prefixlist ORF capability to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouter running ipv4-multicast# capability-orf-prefix-list CAPABILITY-ORF-PREFIX-LIST
```

| CAPABILITY-ORF-PREFIX-LIST values | Description   |
|-----------------------------------|---|
| both                              | Capability to SEND and RECEIVE the ORF to/from this neighbor. |
| send                              | Capability to SEND the ORF to this neighbor.                  |
| receive                           | Capability to RECEIVE the ORF from this neighbor.             |

## unsuppress-map

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouter running ipv4-multicast# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳multicast update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳multicast sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳multicast packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳multicast accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳multicast inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳multicast outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv4-multicast addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv4-multicast addpath  
vrouter running addpath# tx-all-paths true|false
```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv4-multicast addpath  
vrouter running addpath# tx-best-path-per-AS true|false
```

#### Default value

false

### distributed-list

Filter updates to/from this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouters running ipv4-multicast# distributed-list <distributed-list> access-list ACCESS-
↳LIST
```

| <distributed-list> values | Description              |
|---------------------------|--------------------------|
| in                        | Filter incoming updates. |
| out                       | Filter outgoing updates. |

### access-list (mandatory)

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

### maximum-prefix

Maximum number of prefixes to accept from this peer.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast maximum-prefix
```

### maximum (mandatory)

Maximum number of prefix limit.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast maximum-prefix
vrouters running maximum-prefix# maximum <uint32>
```

## threshold

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast maximum-prefix
vrouter running maximum-prefix# threshold <uint8>
```

### Default value

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast maximum-prefix
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast maximum-prefix
vrouter running maximum-prefix# warning-only true|false
```

### Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast nexthop-self
vrouter running nexthop-self# force true|false

```

### Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast as-outbound-update

```

## action

Action to apply for ASNs in outbound updates.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast as-outbound-update
vrouter running as-outbound-update# action ACTION

```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast as-outbound-update
vrouter running as-outbound-update# as-type AS-TYPE

```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

**Default value**

private

**filter-list**

Establish BGP filters.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouter running ipv4-multicast# filter-list <filter-list> access-list <as-path-access-
↳list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list <as-path-access-list>
```

**prefix-list**

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouter running ipv4-multicast# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-
↳NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

### prefix-list-name (mandatory)

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

### route-map

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast
vrouter running ipv4-multicast# route-map <route-map> route-map-name ROUTE-MAP-NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

### route-map-name (mandatory)

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

### default-originate

Originate default route to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast default-originate
```

## route-map

Route-map to specify criteria to originate default.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-multicast default-originate
vrouter running default-originate# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## ipv4-labeled-unicast

IPv4 labeled unicast address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
```

## enabled

Enable or disable IPv4 labeled unicast Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# enabled true|false
```

### Default value

true

## as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# as-override true|false
```

### Default value

false

## maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# maximum-prefix-out <uint32>

```

## send-community

Send Community attribute to this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# send-community SEND-COMMUNITY

```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

## weight

Set default weight for routes from this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# weight <uint16>

```

## allowas-in

Accept as-path with my AS present in it.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# allowas-in ALLOWAS-IN

```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

### attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouters running ipv4-labeled-unicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

### route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouters running ipv4-labeled-unicast# route-reflector-client true|false
```

#### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouters running ipv4-labeled-unicast# route-server-client true|false
```

#### Default value

false

## soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# soft-reconfiguration-inbound true|false
```

### Default value

false

## capability-orf-prefix-list

Advertise prefixlist ORF capability to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# capability-orf-prefix-list CAPABILITY-ORF-PREFIX-
↳LIST
```

| CAPABILITY-ORF-PREFIX-LIST values | Description   |
|-----------------------------------|---|
| both                              | Capability to SEND and RECEIVE the ORF to/from this neighbor. |
| send                              | Capability to SEND the ORF to this neighbor.                  |
| receive                           | Capability to RECEIVE the ORF from this neighbor.             |

## unsuppress-map

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳labeled-unicast update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳labeled-unicast sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳labeled-unicast packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳labeled-unicast accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳labeled-unicast inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳labeled-unicast outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv4-labeled-unicast addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv4-labeled-unicast addpath  
vrouter running addpath# tx-all-paths true|false
```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv4-labeled-unicast addpath  
vrouter running addpath# tx-best-path-per-AS true|false
```

#### Default value

false

## distribute-list

Filter updates to/from this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# distribute-list <distribute-list> access-list_
↳ACCESS-LIST

```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

## access-list (mandatory)

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

## maximum-prefix

Maximum number of prefixes to accept from this peer.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast maximum-prefix

```

## maximum (mandatory)

Maximum number of prefix limit.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast maximum-prefix
vrouter running maximum-prefix# maximum <uint32>

```

## threshold

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast maximum-prefix
vrouter running maximum-prefix# threshold <uint8>
```

### Default value

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast maximum-prefix
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast maximum-prefix
vrouter running maximum-prefix# warning-only true|false
```

### Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast nexthop-self
vrouters running nexthop-self# force true|false
```

### Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast as-outbound-update
```

## action

Action to apply for ASNs in outbound updates.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast as-outbound-update
vrouters running as-outbound-update# action ACTION
```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast as-outbound-update
vrouters running as-outbound-update# as-type AS-TYPE
```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

**Default value**

private

**filter-list**

Establish BGP filters.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# filter-list <filter-list> access-list <as-path-
↳access-list...

```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list <as-path-access-list>
```

**prefix-list**

Filter updates to/from this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# prefix-list <prefix-list> prefix-list-name_
↳PREFIX-LIST-NAME

```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**prefix-list-name (mandatory)**

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

**route-map**

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# route-map <route-map> route-map-name ROUTE-MAP-
↳NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

**route-map-name (mandatory)**

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

**default-originate**

Originate default route to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast default-originate
```

## route-map

Route-map to specify criteria to originate default.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-labeled-unicast default-originate
vrouter running default-originate# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## ipv4-flowspec

IPv4 Flowspec address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-flowspec
```

## enabled

Enable or disable IPv4 Flowspec Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-flowspec
vrouter running ipv4-flowspec# enabled true|false
```

### Default value

true

## route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-flowspec
vrouter running ipv4-flowspec# route-reflector-client true|false
```

### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-flowspec
vrouter running ipv4-flowspec# route-server-client true|false
```

#### Default value

false

### soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-flowspec
vrouter running ipv4-flowspec# soft-reconfiguration-inbound true|false
```

#### Default value

false

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-
↳flowspec update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-
↳flowspec sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳flowspec packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳flowspec accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳flowspec inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳flowspec outbound-ebgp-requires-policy
```

### filter-list

Establish BGP filters.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv4-flowspec  
vrouter running ipv4-flowspec# filter-list <filter-list> access-list <as-path-access-  
↳list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

## access-list (mandatory)

Access list name.

```
access-list <as-path-access-list>
```

## prefix-list

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ ipv4-flowspec
vrouter running ipv4-flowspec# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-
↳ NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

## prefix-list-name (mandatory)

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

## route-map

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ ipv4-flowspec
vrouter running ipv4-flowspec# route-map <route-map> route-map-name ROUTE-MAP-NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

**route-map-name (mandatory)**

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

**ipv4-vpn**

Configure IPv4 VPN address family.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
```

**enabled**

Enable or disable IPv4 VPN Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouter running ipv4-vpn# enabled true|false
```

**Default value**

true

**unsuppress-map**

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouter running ipv4-vpn# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

## as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouter running ipv4-vpn# as-override true|false

```

### Default value

false

## maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouter running ipv4-vpn# maximum-prefix-out <uint32>

```

## send-community

Send Community attribute to this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouter running ipv4-vpn# send-community SEND-COMMUNITY

```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

## weight

Set default weight for routes from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouter running ipv4-vpn# weight <uint16>
```

## allowas-in

Accept as-path with my AS present in it.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouter running ipv4-vpn# allowas-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

## attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouter running ipv4-vpn# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

## route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouter running ipv4-vpn# route-reflector-client true|false
```

**Default value**

false

**route-server-client**

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouter running ipv4-vpn# route-server-client true|false
```

**Default value**

false

**soft-reconfiguration-inbound**

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouter running ipv4-vpn# soft-reconfiguration-inbound true|false
```

**Default value**

false

**update-group-id (state only)**

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-
↳vpn update-group-id
```

**sub-group-id (state only)**

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-
↳vpn sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳vpn packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳vpn accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳vpn inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv4-  
↳vpn outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family _  
↳ipv4-vpn addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn addpath
vrouters running addpath# tx-all-paths true|false

```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn addpath
vrouters running addpath# tx-best-path-per-AS true|false

```

#### Default value

false

### distribute-list

Filter updates to/from this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouters running ipv4-vpn# distribute-list <distribute-list> access-list ACCESS-LIST

```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

**maximum-prefix**

Maximum number of prefixes to accept from this peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn maximum-prefix
```

**maximum (mandatory)**

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn maximum-prefix
vrouter running maximum-prefix# maximum <uint32>
```

**threshold**

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn maximum-prefix
vrouter running maximum-prefix# threshold <uint8>
```

**Default value**

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn maximum-prefix
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn maximum-prefix
vrouter running maximum-prefix# warning-only true|false
```

## Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn nexthop-self
vrouter running nexthop-self# force true|false
```

## Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn as-outbound-update
```

### action

Action to apply for ASNs in outbound updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn as-outbound-update
vrouter running as-outbound-update# action ACTION
```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn as-outbound-update
vrouter running as-outbound-update# as-type AS-TYPE
```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

### Default value

private

### filter-list

Establish BGP filters.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouter running ipv4-vpn# filter-list <filter-list> access-list <as-path-access-list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

### access-list (mandatory)

Access list name.

```
access-list <as-path-access-list>
```

### prefix-list

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouter running ipv4-vpn# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

### prefix-list-name (mandatory)

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

## route-map

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv4-vpn
vrouter running ipv4-vpn# route-map <route-map> route-map-name ROUTE-MAP-NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

## route-map-name (mandatory)

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

## ipv6-unicast

IPv6 unicast address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
```

## enabled

Enable or disable IPv6 unicast Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouter running ipv6-unicast# enabled true|false
```

## Default value

true

### nexthop-local-unchanged

If true, leave link-local nexthop unchanged for this peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouter running ipv6-unicast# nexthop-local-unchanged true|false
```

#### Default value

false

### as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouter running ipv6-unicast# as-override true|false
```

#### Default value

false

### maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouter running ipv6-unicast# maximum-prefix-out <uint32>
```

### send-community

Send Community attribute to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouter running ipv6-unicast# send-community SEND-COMMUNITY
```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

### weight

Set default weight for routes from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouter running ipv6-unicast# weight <uint16>
```

### allowas-in

Accept as-path with my AS present in it.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouter running ipv6-unicast# allowas-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

### attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouter running ipv6-unicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

### route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouters running ipv6-unicast# route-reflector-client true|false
```

#### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouters running ipv6-unicast# route-server-client true|false
```

#### Default value

false

### soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouters running ipv6-unicast# soft-reconfiguration-inbound true|false
```

#### Default value

false

### capability-orf-prefix-list

Advertise prefixlist ORF capability to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouter running ipv6-unicast# capability-orf-prefix-list CAPABILITY-ORF-PREFIX-LIST
```

| CAPABILITY-ORF-PREFIX-LIST values | Description   |
|-----------------------------------|---|
| both                              | Capability to SEND and RECEIVE the ORF to/from this neighbor. |
| send                              | Capability to SEND the ORF to this neighbor.                  |
| receive                           | Capability to RECEIVE the ORF from this neighbor.             |

### unsuppress-map

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouter running ipv6-unicast# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-
↳unicast update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-
↳unicast sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳unicast packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳unicast accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳unicast inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳unicast outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family _  
↳ipv6-unicast addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast addpath
vrouters running addpath# tx-all-paths true|false

```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast addpath
vrouters running addpath# tx-best-path-per-AS true|false

```

#### Default value

false

### distribute-list

Filter updates to/from this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouters running ipv6-unicast# distribute-list <distribute-list> access-list ACCESS-LIST

```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

**maximum-prefix**

Maximum number of prefixes to accept from this peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast maximum-prefix
```

**maximum (mandatory)**

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast maximum-prefix
vrouter running maximum-prefix# maximum <uint32>
```

**threshold**

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast maximum-prefix
vrouter running maximum-prefix# threshold <uint8>
```

**Default value**

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast maximum-prefix
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast maximum-prefix
vrouter running maximum-prefix# warning-only true|false
```

## Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast nexthop-self
vrouter running nexthop-self# force true|false
```

## Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast as-outbound-update
```

### action

Action to apply for ASNs in outbound updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast as-outbound-update
vrouter running as-outbound-update# action ACTION
```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast as-outbound-update
vrouter running as-outbound-update# as-type AS-TYPE
```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

### Default value

private

## filter-list

Establish BGP filters.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouter running ipv6-unicast# filter-list <filter-list> access-list <as-path-access-
↳list...

```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

## access-list (mandatory)

Access list name.

```
access-list <as-path-access-list>
```

## prefix-list

Filter updates to/from this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouter running ipv6-unicast# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-
↳NAME

```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

## prefix-list-name (mandatory)

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

## route-map

Apply route map to this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast
vrouter running ipv6-unicast# route-map <route-map> route-map-name ROUTE-MAP-NAME

```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

## route-map-name (mandatory)

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

## default-originate

Originate default route to this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast default-originate

```

## route-map

Route-map to specify criteria to originate default.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-unicast default-originate
vrouter running default-originate# route-map ROUTE-MAP

```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## ipv6-multicast

IPv6 multicast address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
```

### enabled

Enable or disable IPv6 multicast Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# enabled true|false
```

### Default value

true

### as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# as-override true|false
```

### Default value

false

### maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# maximum-prefix-out <uint32>
```

## send-community

Send Community attribute to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# send-community SEND-COMMUNITY
```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

## weight

Set default weight for routes from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# weight <uint16>
```

## allowas-in

Accept as-path with my AS present in it.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# allowas-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

## attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

## route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# route-reflector-client true|false
```

### Default value

false

## route-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# route-server-client true|false
```

### Default value

false

## soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# soft-reconfiguration-inbound true|false
```

### Default value

false

## capability-orf-prefix-list

Advertise prefixlist ORF capability to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# capability-orf-prefix-list CAPABILITY-ORF-PREFIX-LIST
```

| CAPABILITY-ORF-PREFIX-LIST values | Description   |
|-----------------------------------|---|
| both                              | Capability to SEND and RECEIVE the ORF to/from this neighbor. |
| send                              | Capability to SEND the ORF to this neighbor.                  |
| receive                           | Capability to RECEIVE the ORF from this neighbor.             |

## unsuppress-map

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳multicast update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳multicast sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳multicast packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳multicast accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳multicast inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳multicast outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv6-multicast addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv6-multicast addpath  
vrouter running addpath# tx-all-paths true|false
```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv6-multicast addpath  
vrouter running addpath# tx-best-path-per-AS true|false
```

#### Default value

false

## distribute-list

Filter updates to/from this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouters running ipv6-multicast# distribute-list <distribute-list> access-list ACCESS-
↳LIST

```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

## access-list (mandatory)

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

## maximum-prefix

Maximum number of prefixes to accept from this peer.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast maximum-prefix

```

## maximum (mandatory)

Maximum number of prefix limit.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast maximum-prefix
vrouters running maximum-prefix# maximum <uint32>

```

## threshold

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast maximum-prefix
vrouter running maximum-prefix# threshold <uint8>
```

### Default value

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast maximum-prefix
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast maximum-prefix
vrouter running maximum-prefix# warning-only true|false
```

### Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast nexthop-self
vrouter running nexthop-self# force true|false

```

### Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast as-outbound-update

```

## action

Action to apply for ASNs in outbound updates.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast as-outbound-update
vrouter running as-outbound-update# action ACTION

```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast as-outbound-update
vrouter running as-outbound-update# as-type AS-TYPE

```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

**Default value**

private

**filter-list**

Establish BGP filters.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# filter-list <filter-list> access-list <as-path-access-
↳list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list <as-path-access-list>
```

**prefix-list**

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-
↳NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**prefix-list-name (mandatory)**

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

**route-map**

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast
vrouter running ipv6-multicast# route-map <route-map> route-map-name ROUTE-MAP-NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

**route-map-name (mandatory)**

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

**default-originate**

Originate default route to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast default-originate
```

## route-map

Route-map to specify criteria to originate default.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-multicast default-originate
vrouter running default-originate# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## ipv6-labeled-unicast

IPv6 labeled unicast address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
```

## enabled

Enable or disable IPv6 labeled unicast Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouter running ipv6-labeled-unicast# enabled true|false
```

### Default value

true

## as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouter running ipv6-labeled-unicast# as-override true|false
```

### Default value

false

## maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouters running ipv6-labeled-unicast# maximum-prefix-out <uint32>

```

## send-community

Send Community attribute to this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouters running ipv6-labeled-unicast# send-community SEND-COMMUNITY

```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

## weight

Set default weight for routes from this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouters running ipv6-labeled-unicast# weight <uint16>

```

## allowas-in

Accept as-path with my AS present in it.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouters running ipv6-labeled-unicast# allowas-in ALLOWAS-IN

```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

### attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouters running ipv6-labeled-unicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

### route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouters running ipv6-labeled-unicast# route-reflector-client true|false
```

#### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouters running ipv6-labeled-unicast# route-server-client true|false
```

#### Default value

false

## soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouter running ipv6-labeled-unicast# soft-reconfiguration-inbound true|false
```

### Default value

false

## capability-orf-prefix-list

Advertise prefixlist ORF capability to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouter running ipv6-labeled-unicast# capability-orf-prefix-list CAPABILITY-ORF-PREFIX-
↳LIST
```

| CAPABILITY-ORF-PREFIX-LIST values | Description   |
|-----------------------------------|---|
| both                              | Capability to SEND and RECEIVE the ORF to/from this neighbor. |
| send                              | Capability to SEND the ORF to this neighbor.                  |
| receive                           | Capability to RECEIVE the ORF from this neighbor.             |

## unsuppress-map

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouter running ipv6-labeled-unicast# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳labeled-unicast update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳labeled-unicast sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳labeled-unicast packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳labeled-unicast accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳labeled-unicast inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳labeled-unicast outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv6-labeled-unicast addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv6-labeled-unicast addpath  
vrouter running addpath# tx-all-paths true|false
```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv6-labeled-unicast addpath  
vrouter running addpath# tx-best-path-per-AS true|false
```

#### Default value

false

### distributed-list

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouter running ipv6-labeled-unicast# distribute-list <distribute-list> access-list_
↳ACCESS-LIST
```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

### access-list (mandatory)

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

### maximum-prefix

Maximum number of prefixes to accept from this peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast maximum-prefix
```

### maximum (mandatory)

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast maximum-prefix
vrouter running maximum-prefix# maximum <uint32>
```

## threshold

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast maximum-prefix
vrouter running maximum-prefix# threshold <uint8>
```

### Default value

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast maximum-prefix
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast maximum-prefix
vrouter running maximum-prefix# warning-only true|false
```

### Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast nexthop-self
vrouter running nexthop-self# force true|false

```

### Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast as-outbound-update

```

## action

Action to apply for ASNs in outbound updates.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast as-outbound-update
vrouter running as-outbound-update# action ACTION

```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast as-outbound-update
vrouter running as-outbound-update# as-type AS-TYPE

```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

**Default value**

private

**filter-list**

Establish BGP filters.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouters running ipv6-labeled-unicast# filter-list <filter-list> access-list <as-path-
↳access-list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list <as-path-access-list>
```

**prefix-list**

Filter updates to/from this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouters running ipv6-labeled-unicast# prefix-list <prefix-list> prefix-list-name_
↳PREFIX-LIST-NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**prefix-list-name (mandatory)**

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

**route-map**

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast
vrouter running ipv6-labeled-unicast# route-map <route-map> route-map-name ROUTE-MAP-
↳NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

**route-map-name (mandatory)**

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

**default-originate**

Originate default route to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast default-originate
```

## route-map

Route-map to specify criteria to originate default.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-labeled-unicast default-originate
vrouter running default-originate# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## ipv6-flowspec

IPv6 Flowspec address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-flowspec
```

## enabled

Enable or disable IPv6 Flowspec Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-flowspec
vrouter running ipv6-flowspec# enabled true|false
```

### Default value

true

## route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-flowspec
vrouter running ipv6-flowspec# route-reflector-client true|false
```

### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-flowspec
vrouter running ipv6-flowspec# route-server-client true|false
```

#### Default value

false

### soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-flowspec
vrouter running ipv6-flowspec# soft-reconfiguration-inbound true|false
```

#### Default value

false

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-
↳flowspec update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-
↳flowspec sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳flowspec packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳flowspec accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳flowspec inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳flowspec outbound-ebgp-requires-policy
```

### filter-list

Establish BGP filters.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv6-flowspec  
vrouter running ipv6-flowspec# filter-list <filter-list> access-list <as-path-access-  
↳list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

## access-list (mandatory)

Access list name.

```
access-list <as-path-access-list>
```

## prefix-list

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ ipv6-flowspec
vrouter running ipv6-flowspec# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-
↳ NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

## prefix-list-name (mandatory)

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

## route-map

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ ipv6-flowspec
vrouter running ipv6-flowspec# route-map <route-map> route-map-name ROUTE-MAP-NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

### route-map-name (mandatory)

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

### ipv6-vpn

Configure IPv6 VPN address family.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv6-vpn
```

### enabled

Enable or disable IPv6 VPN Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv6-vpn  
vrouter running ipv6-vpn# enabled true|false
```

### Default value

true

### unsuppress-map

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_  
↳ipv6-vpn  
vrouter running ipv6-vpn# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

## as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn
vrouters running ipv6-vpn# as-override true|false

```

### Default value

false

## maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn
vrouters running ipv6-vpn# maximum-prefix-out <uint32>

```

## send-community

Send Community attribute to this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn
vrouters running ipv6-vpn# send-community SEND-COMMUNITY

```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

## weight

Set default weight for routes from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn
vrouter running ipv6-vpn# weight <uint16>
```

## allowas-in

Accept as-path with my AS present in it.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn
vrouter running ipv6-vpn# allowas-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

## attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn
vrouter running ipv6-vpn# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

## route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn
vrouter running ipv6-vpn# route-reflector-client true|false
```

**Default value**

false

**route-server-client**

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family _  
↳ ipv6-vpn  
vrouter running ipv6-vpn# route-server-client true|false
```

**Default value**

false

**soft-reconfiguration-inbound**

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family _  
↳ ipv6-vpn  
vrouter running ipv6-vpn# soft-reconfiguration-inbound true|false
```

**Default value**

false

**update-group-id (state only)**

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳ vpn update-group-id
```

**sub-group-id (state only)**

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳ vpn sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳vpn packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳vpn accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳vpn inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family ipv6-  
↳vpn outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family _  
↳ipv6-vpn addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn addpath
vrouter running addpath# tx-all-paths true|false

```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn addpath
vrouter running addpath# tx-best-path-per-AS true|false

```

#### Default value

false

### distribute-list

Filter updates to/from this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn
vrouter running ipv6-vpn# distribute-list <distribute-list> access-list ACCESS-LIST

```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

**maximum-prefix**

Maximum number of prefixes to accept from this peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn maximum-prefix
```

**maximum (mandatory)**

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn maximum-prefix
vrouter running maximum-prefix# maximum <uint32>
```

**threshold**

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn maximum-prefix
vrouter running maximum-prefix# threshold <uint8>
```

**Default value**

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn maximum-prefix
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn maximum-prefix
vrouter running maximum-prefix# warning-only true|false
```

## Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn nexthop-self
vrouter running nexthop-self# force true|false
```

## Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn as-outbound-update
```

### action

Action to apply for ASNs in outbound updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn as-outbound-update
vrouter running as-outbound-update# action ACTION
```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn as-outbound-update
vrouter running as-outbound-update# as-type AS-TYPE
```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

### Default value

private

### filter-list

Establish BGP filters.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn
vrouter running ipv6-vpn# filter-list <filter-list> access-list <as-path-access-list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

### access-list (mandatory)

Access list name.

```
access-list <as-path-access-list>
```

### prefix-list

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn
vrouter running ipv6-vpn# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

### prefix-list-name (mandatory)

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

## route-map

Apply route map to this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ipv6-vpn
vrouter running ipv6-vpn# route-map <route-map> route-map-name ROUTE-MAP-NAME

```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

## route-map-name (mandatory)

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

## l2vpn-evpn

Configure L2VPN EVPN address family.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳l2vpn-evpn

```

## enabled

Enable or disable L2VPN EVPN Address Family for this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳l2vpn-evpn
vrouter running l2vpn-evpn# enabled true|false

```

## Default value

true

### nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳l2vpn-evpn
vrouter running l2vpn-evpn# nexthop-self true|false
```

#### Default value

false

### route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳l2vpn-evpn
vrouter running l2vpn-evpn# route-reflector-client true|false
```

#### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳l2vpn-evpn
vrouter running l2vpn-evpn# route-server-client true|false
```

#### Default value

false

### soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳l2vpn-evpn
vrouter running l2vpn-evpn# soft-reconfiguration-inbound true|false
```

#### Default value

false

### allowas-in

Accept as-path with my AS present in it.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳l2vpn-evpn
vrouter running l2vpn-evpn# allowas-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

### attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳l2vpn-evpn
vrouter running l2vpn-evpn# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family l2vpn-
↳evpn update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family l2vpn-  
↳evpn sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family l2vpn-  
↳evpn packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family l2vpn-  
↳evpn accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family l2vpn-  
↳evpn inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> address-family l2vpn-  
↳evpn outbound-ebgp-requires-policy
```

## route-map

Apply route map to this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor-group <string> address-family_
↳ l2vpn-evpn
vrouters running l2vpn-evpn# route-map <route-map> route-map-name ROUTE-MAP-NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

## route-map-name (mandatory)

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

## connections (state only)

Established/dropped connections statistics.

## established (state only)

Number of established connections.

```
vrouters> show state vrf <vrf> routing bgp neighbor-group <string> connections_
↳ established
```

## dropped (state only)

Number of dropped connections.

```
vrouters> show state vrf <vrf> routing bgp neighbor-group <string> connections dropped
```

**local-host (state only)**

Local host data.

**name (state only)**

Local host name.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> local-host name
```

**port (state only)**

Local host port.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> local-host port
```

**remote-host (state only)**

Remote host data.

**name (state only)**

Remote host name.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> remote-host name
```

**port (state only)**

Remote host port.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> remote-host port
```

### nexthop (state only)

Nexthop data.

### address (state only)

Nexthop IP address.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> nexthop address
```

### global-address (state only)

Nexthop global IP address.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> nexthop global-  
↪address
```

### local-address (state only)

Nexthop local IP address.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> nexthop local-address
```

### thread (state only)

Read/Write thread.

### read-enabled (state only)

Read thread status.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> thread read-enabled
```

### write-enabled (state only)

Write thread status.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> thread write-enabled
```

### message-statistics (state only)

Neighbor messages statistics.

### packet-wait-process (state only)

Number of packets waiting to be processed.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳packet-wait-process
```

### packet-wait-written (state only)

Number of packets waiting to be written.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳packet-wait-written
```

### open-sent (state only)

BGP open messages sent.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳open-sent
```

### opens-received (state only)

BGP open messages received.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳opens-received
```

### notifications-sent (state only)

Notifications messages sent.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳notifications-sent
```

### notifications-received (state only)

Notification messages received.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳notifications-received
```

### updates-sent (state only)

Update messages sent.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳updates-sent
```

### updates-received (state only)

Update messages received.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳updates-received
```

### keepalives-sent (state only)

Keepalive messages sent.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳keepalives-sent
```

### keepalives-received (state only)

Keepalive messages received.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳keepalives-received
```

### route-refresh-sent (state only)

Route refresh messages sent.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳route-refresh-sent
```

### route-refresh-received (state only)

Route refresh messages received.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳route-refresh-received
```

### capability-sent (state only)

Capability messages sent.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳capability-sent
```

### capability-received (state only)

Capability messages received.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳capability-received
```

### total-sent (state only)

Total messages sent.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳total-sent
```

### total-received (state only)

Total messages received.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> message-statistics_
↳total-received
```

### neighbor

List of BGP neighbors configured on the local system, uniquely identified by peer IPv[46] address.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>
```

| <neighbor> values | Description      |
|-------------------|------------------|
| <A.B.C.D>         | An IPv4 address. |
| <X:X::X:X>        | An IPv6 address. |

### neighbor-group

Peer group name.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>
vrouter running neighbor <neighbor># neighbor-group <neighbor-group>
```

### interface

Name of the interface.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>
vrouter running neighbor <neighbor># interface INTERFACE
```

|           |                    |
|-----------|--------------------|
| INTERFACE | An interface name. |
|-----------|--------------------|

## port

TCP port number.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>
vrouter running neighbor <neighbor># port PORT
```

|      |   |
|------|---|
| PORT | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|------|---|

## remote-as

Remote AS number.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>
vrouter running neighbor <neighbor># remote-as REMOTE-AS
```

| REMOTE-AS values | Description        |
|------------------|--------------------|
| <uint32>         | No description.    |
| external         | External BGP peer. |
| internal         | Internal BGP peer. |

## capability

Advertise capability to the peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>
vrouter running neighbor <neighbor># capability CAPABILITY
```

| CAPABILITY values | Description   |
|-------------------|---|
| dynamic           | Advertise dynamic capability to this neighbor.      |
| extended-nextthop | Advertise extended nextthop capability to the peer. |

## capability-negotiate

If true, perform capability negotiation.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>
vrouter running neighbor <neighbor># capability-negotiate true|false
```

## Default value

true

### ebgp-multihop

Allow EBGp neighbors not on directly connected networks.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>  
vrouter running neighbor <neighbor># ebgp-multihop <uint8>
```

### enforce-first-as

If true, enforce the first AS for EBGp routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>  
vrouter running neighbor <neighbor># enforce-first-as true|false
```

#### Default value

false

### enforce-multihop

If true, enforce EBGp neighbors perform multihop.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>  
vrouter running neighbor <neighbor># enforce-multihop true|false
```

#### Default value

false

### neighbor-description

Neighbor specific description: up to 80 characters describing this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>  
vrouter running neighbor <neighbor># neighbor-description <string>
```

### override-capability

If true, override capability negotiation result.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>  
vrouter running neighbor <neighbor># override-capability true|false
```

#### Default value

false

### passive

If true, don't send open messages to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>  
vrouter running neighbor <neighbor># passive true|false
```

#### Default value

false

### password

Set a password.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>  
vrouter running neighbor <neighbor># password <string>
```

### solo

If true, solo peer - part of its own update group.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>  
vrouter running neighbor <neighbor># solo true|false
```

#### Default value

false

### strict-capability-match

Enable or disable strict capability negotiation match.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>
vrouter running neighbor <neighbor># strict-capability-match true|false
```

#### Default value

false

### track

A tracker name defined in the tracker context, or the BGP internal BFD tracker (bfd keyword). If a tracker name is used, when the tracked address is reachable, the neighbor or neighbor group is considered as valid, else it is disabled. If the BGP internal BFD tracker is used, it works the same way, but this neighbor address is automatically tracked. The check-control-plane-failure option is available only when the BGP internal BFD tracker is used.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>
vrouter running neighbor <neighbor># track TRACK
```

| TRACK values   | Description      |
|----------------|------------------|
| <tracker-name> | An tracker name. |
| <identityref>  | No description.  |

### check-control-plane-failure

Link data-plane status with BGP control-plane. This option is available only if the BGP internal BFD tracker is selected, i.e the ‘track bfd’ option is set.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>
vrouter running neighbor <neighbor># check-control-plane-failure true|false
```

### ttl-security-hops

Specify the maximum number of hops to the BGP peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>
vrouter running neighbor <neighbor># ttl-security-hops <uint8>
```

## update-source

Source of routing updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>
vrouter running neighbor <neighbor># update-source UPDATE-SOURCE
```

| UPDATE-SOURCE values | Description        |
|----------------------|--------------------|
| <A.B.C.D>            | An IPv4 address.   |
| <X:X::X:X>           | An IPv6 address.   |
| <ifname>             | An interface name. |

## sender-as-path-loop-detection

Detect the sender side AS path loops and filter the bad routes before they are sent.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor>
vrouter running neighbor <neighbor># sender-as-path-loop-detection true|false
```

### Default value

false

## remote-neighbor-group (state only)

Remote neighbor group.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> remote-neighbor-group
```

## remote-router-id (state only)

Remote router identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> remote-router-id
```

### state (state only)

BGP router status.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> state
```

### min-time-btwn-advertisement (state only)

Minimum time between advertisement runs in milliseconds.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> min-time-btwn-  
↵advertisement
```

### last-reset (state only)

Last reset.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> last-reset
```

### bgp-connection (state only)

BGP connection type.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> bgp-connection
```

### connect-retry-timer (state only)

BGP connect retry timer in seconds.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> connect-retry-timer
```

### estimated-round-trip-time (state only)

Estimated round trip time in milliseconds.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> estimated-round-trip-time
```

## local-as

Specify a local-as number.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> local-as
```

## as-number (mandatory)

AS number used as local AS.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> local-as  
vrouter running local-as# as-number <uint32>
```

## no-prepend

If true, do not prepend local-as to updates from ebgp peers.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> local-as  
vrouter running local-as# no-prepend true|false
```

### Default value

false

## replace-as

If true, do not prepend local-as to updates from ibgp peers.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> local-as  
vrouter running local-as# replace-as true|false
```

### Default value

false

## shutdown

Administratively shut down this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> shutdown
```

## message

Shutdown message.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> shutdown
vrouter running shutdown# message <string>
```

## timers

Config parameters related to timers associated with the BGP peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> timers
```

## advertisement-interval

Minimum time which must elapse between subsequent UPDATE messages relating to a common set of NLRI being transmitted to a peer. This timer is referred to as MinRouteAdvertisementIntervalTimer by RFC 4721 and serves to reduce the number of UPDATE messages transmitted when a particular set of NLRI exhibit instability. A change of this value will be taken into account for new sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> timers
vrouter running timers# advertisement-interval <uint16>
```

## connect-retry

Time interval in seconds between attempts to establish a session with the peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> timers
vrouter running timers# connect-retry <uint16>
```

## keepalive-interval

Time interval in seconds between transmission of keepalive messages to the neighbor. Typically set to 1/3 the hold-time. A change of this value will be taken into account for new sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> timers
vrouter running timers# keepalive-interval <uint16>
```

## hold-time

Time interval in seconds that a BGP session will be considered active in the absence of keepalive or other messages from the peer. The hold-time is typically set to 3x the keepalive-interval.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> timers
vrouter running timers# hold-time <uint16>
```

## address-family

Address-families associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family
```

## ipv4-unicast

IPv4 unicast address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
```

## enabled

Enable or disable IPv4 unicast Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouter running ipv4-unicast# enabled true|false
```

## Default value

true

## as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouter running ipv4-unicast# as-override true|false
```

## Default value

false

### maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouter running ipv4-unicast# maximum-prefix-out <uint32>
```

### send-community

Send Community attribute to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouter running ipv4-unicast# send-community SEND-COMMUNITY
```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

### weight

Set default weight for routes from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouter running ipv4-unicast# weight <uint16>
```

## allowas-in

Accept as-path with my AS present in it.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouters running ipv4-unicast# allowas-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

## attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouters running ipv4-unicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

## route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouters running ipv4-unicast# route-reflector-client true|false
```

### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouter running ipv4-unicast# route-server-client true|false
```

#### Default value

false

### soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouter running ipv4-unicast# soft-reconfiguration-inbound true|false
```

#### Default value

false

### capability-orf-prefix-list

Advertise prefixlist ORF capability to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouter running ipv4-unicast# capability-orf-prefix-list CAPABILITY-ORF-PREFIX-LIST
```

| CAPABILITY-ORF-PREFIX-LIST values | Description   |
|-----------------------------------|---|
| both                              | Capability to SEND and RECEIVE the ORF to/from this neighbor. |
| send                              | Capability to SEND the ORF to this neighbor.                  |
| receive                           | Capability to RECEIVE the ORF from this neighbor.             |

## unsuppress-map

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳unicast  
vrouter running ipv4-unicast# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

## update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳unicast update-group-id
```

## sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳unicast sub-group-id
```

## packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳unicast packet-queue-length
```

## accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳unicast accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast addpath
vrouter running addpath# tx-all-paths true|false
```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast addpath
vrouter running addpath# tx-best-path-per-AS true|false
```

#### Default value

false

### distributed-list

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouter running ipv4-unicast# distribute-list <distribute-list> access-list ACCESS-LIST
```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

### access-list (mandatory)

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

### maximum-prefix

Maximum number of prefixes to accept from this peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast maximum-prefix
```

### maximum (mandatory)

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast maximum-prefix
vrouter running maximum-prefix# maximum <uint32>
```

## threshold

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳unicast maximum-prefix  
vrouter running maximum-prefix# threshold <uint8>
```

### Default value

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳unicast maximum-prefix  
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳unicast maximum-prefix  
vrouter running maximum-prefix# warning-only true|false
```

### Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳unicast nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast nexthop-self
vrouters running nexthop-self# force true|false
```

### Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast as-outbound-update
```

## action

Action to apply for ASNs in outbound updates.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast as-outbound-update
vrouters running as-outbound-update# action ACTION
```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast as-outbound-update
vrouters running as-outbound-update# as-type AS-TYPE
```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

**Default value**

private

**filter-list**

Establish BGP filters.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouters running ipv4-unicast# filter-list <filter-list> access-list <as-path-access-
↳list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list <as-path-access-list>
```

**prefix-list**

Filter updates to/from this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouters running ipv4-unicast# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-
↳NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**prefix-list-name (mandatory)**

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

**route-map**

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast
vrouter running ipv4-unicast# route-map <route-map> route-map-name ROUTE-MAP-NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

**route-map-name (mandatory)**

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

**default-originate**

Originate default route to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast default-originate
```

## route-map

Route-map to specify criteria to originate default.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳unicast default-originate
vrouter running default-originate# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## ipv4-multicast

IPv4 multicast address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
```

## enabled

Enable or disable IPv4 multicast Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouter running ipv4-multicast# enabled true|false
```

### Default value

true

## as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouter running ipv4-multicast# as-override true|false
```

### Default value

false

## maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouters running ipv4-multicast# maximum-prefix-out <uint32>
```

## send-community

Send Community attribute to this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouters running ipv4-multicast# send-community SEND-COMMUNITY
```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

## weight

Set default weight for routes from this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouters running ipv4-multicast# weight <uint16>
```

## allowas-in

Accept as-path with my AS present in it.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouters running ipv4-multicast# allowas-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

### attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouters running ipv4-multicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

### route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouters running ipv4-multicast# route-reflector-client true|false
```

#### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouters running ipv4-multicast# route-server-client true|false
```

#### Default value

false

## soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouter running ipv4-multicast# soft-reconfiguration-inbound true|false
```

### Default value

false

## capability-orf-prefix-list

Advertise prefixlist ORF capability to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouter running ipv4-multicast# capability-orf-prefix-list CAPABILITY-ORF-PREFIX-LIST
```

| CAPABILITY-ORF-PREFIX-LIST values | Description   |
|-----------------------------------|---|
| both                              | Capability to SEND and RECEIVE the ORF to/from this neighbor. |
| send                              | Capability to SEND the ORF to this neighbor.                  |
| receive                           | Capability to RECEIVE the ORF from this neighbor.             |

## unsuppress-map

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouter running ipv4-multicast# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳multicast update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳multicast sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳multicast packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳multicast accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳multicast inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳multicast outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳multicast addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳multicast addpath  
vrouter running addpath# tx-all-paths true|false
```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳multicast addpath  
vrouter running addpath# tx-best-path-per-AS true|false
```

#### Default value

false

## distribute-list

Filter updates to/from this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouters running ipv4-multicast# distribute-list <distribute-list> access-list ACCESS-
↳LIST

```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

## access-list (mandatory)

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

## maximum-prefix

Maximum number of prefixes to accept from this peer.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast maximum-prefix

```

## maximum (mandatory)

Maximum number of prefix limit.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast maximum-prefix
vrouters running maximum-prefix# maximum <uint32>

```

## threshold

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳multicast maximum-prefix  
vrouter running maximum-prefix# threshold <uint8>
```

### Default value

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳multicast maximum-prefix  
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳multicast maximum-prefix  
vrouter running maximum-prefix# warning-only true|false
```

### Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳multicast nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast nexthop-self
vrouters running nexthop-self# force true|false
```

### Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast as-outbound-update
```

## action

Action to apply for ASNs in outbound updates.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast as-outbound-update
vrouters running as-outbound-update# action ACTION
```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast as-outbound-update
vrouters running as-outbound-update# as-type AS-TYPE
```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

**Default value**

private

**filter-list**

Establish BGP filters.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouters running ipv4-multicast# filter-list <filter-list> access-list <as-path-access-
↳list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list <as-path-access-list>
```

**prefix-list**

Filter updates to/from this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouters running ipv4-multicast# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-
↳NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

### prefix-list-name (mandatory)

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

### route-map

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast
vrouter running ipv4-multicast# route-map <route-map> route-map-name ROUTE-MAP-NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

### route-map-name (mandatory)

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

### default-originate

Originate default route to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳multicast default-originate
```

## route-map

Route-map to specify criteria to originate default.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳multicast default-originate  
vrouter running default-originate# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## ipv4-labeled-unicast

IPv4 labeled unicast address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast
```

## enabled

Enable or disable IPv4 labeled unicast Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast  
vrouter running ipv4-labeled-unicast# enabled true|false
```

### Default value

true

## as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast  
vrouter running ipv4-labeled-unicast# as-override true|false
```

### Default value

false

## maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast
vrouters running ipv4-labeled-unicast# maximum-prefix-out <uint32>
```

## send-community

Send Community attribute to this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast
vrouters running ipv4-labeled-unicast# send-community SEND-COMMUNITY
```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

## weight

Set default weight for routes from this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast
vrouters running ipv4-labeled-unicast# weight <uint16>
```

## allowas-in

Accept as-path with my AS present in it.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast
vrouters running ipv4-labeled-unicast# allowas-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

### attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast
vrouters running ipv4-labeled-unicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

### route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast
vrouters running ipv4-labeled-unicast# route-reflector-client true|false
```

#### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast
vrouters running ipv4-labeled-unicast# route-server-client true|false
```

#### Default value

false

## soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast
vrouter running ipv4-labeled-unicast# soft-reconfiguration-inbound true|false
```

### Default value

false

## capability-orf-prefix-list

Advertise prefixlist ORF capability to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast
vrouter running ipv4-labeled-unicast# capability-orf-prefix-list CAPABILITY-ORF-PREFIX-
↳LIST
```

| CAPABILITY-ORF-PREFIX-LIST values | Description   |
|-----------------------------------|---|
| both                              | Capability to SEND and RECEIVE the ORF to/from this neighbor. |
| send                              | Capability to SEND the ORF to this neighbor.                  |
| receive                           | Capability to RECEIVE the ORF from this neighbor.             |

## unsuppress-map

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast
vrouter running ipv4-labeled-unicast# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast addpath  
vrouter running addpath# tx-all-paths true|false
```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast addpath  
vrouter running addpath# tx-best-path-per-AS true|false
```

#### Default value

false

### distributed-list

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast
vrouter running ipv4-labeled-unicast# distribute-list <distribute-list> access-list_
↳ACCESS-LIST
```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

### access-list (mandatory)

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

### maximum-prefix

Maximum number of prefixes to accept from this peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast maximum-prefix
```

### maximum (mandatory)

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast maximum-prefix
vrouter running maximum-prefix# maximum <uint32>
```

## threshold

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast maximum-prefix  
vrouter running maximum-prefix# threshold <uint8>
```

### Default value

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast maximum-prefix  
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast maximum-prefix  
vrouter running maximum-prefix# warning-only true|false
```

### Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast nexthop-self
vrouter running nexthop-self# force true|false

```

### Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast as-outbound-update

```

## action

Action to apply for ASNs in outbound updates.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast as-outbound-update
vrouter running as-outbound-update# action ACTION

```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast as-outbound-update
vrouter running as-outbound-update# as-type AS-TYPE

```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

**Default value**

private

**filter-list**

Establish BGP filters.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast
vrouters running ipv4-labeled-unicast# filter-list <filter-list> access-list <as-path-
↳access-list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list <as-path-access-list>
```

**prefix-list**

Filter updates to/from this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast
vrouters running ipv4-labeled-unicast# prefix-list <prefix-list> prefix-list-name.
↳PREFIX-LIST-NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**prefix-list-name (mandatory)**

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

**route-map**

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast
vrouter running ipv4-labeled-unicast# route-map <route-map> route-map-name ROUTE-MAP-
↳NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

**route-map-name (mandatory)**

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

**default-originate**

Originate default route to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳labeled-unicast default-originate
```

## route-map

Route-map to specify criteria to originate default.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳labeled-unicast default-originate  
vrouter running default-originate# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## ipv4-flowspec

IPv4 Flowspec address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳flowspec
```

## enabled

Enable or disable IPv4 Flowspec Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳flowspec  
vrouter running ipv4-flowspec# enabled true|false
```

### Default value

true

## route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳flowspec  
vrouter running ipv4-flowspec# route-reflector-client true|false
```

### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳flowspec  
vrouter running ipv4-flowspec# route-server-client true|false
```

#### Default value

false

### soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳flowspec  
vrouter running ipv4-flowspec# soft-reconfiguration-inbound true|false
```

#### Default value

false

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳flowspec update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳flowspec sub-group-id
```

**packet-queue-length (state only)**

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳flowspec packet-queue-length
```

**accepted-prefix (state only)**

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳flowspec accepted-prefix
```

**inbound-ebgp-requires-policy (state only)**

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳flowspec inbound-ebgp-requires-policy
```

**outbound-ebgp-requires-policy (state only)**

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳flowspec outbound-ebgp-requires-policy
```

**filter-list**

Establish BGP filters.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳flowspec
vrouter running ipv4-flowspec# filter-list <filter-list> access-list <as-path-access-
↳list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

## access-list (mandatory)

Access list name.

```
access-list <as-path-access-list>
```

## prefix-list

Filter updates to/from this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳flowspec
vrouters running ipv4-flowspec# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-
↳NAME

```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

## prefix-list-name (mandatory)

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

## route-map

Apply route map to this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳flowspec
vrouters running ipv4-flowspec# route-map <route-map> route-map-name ROUTE-MAP-NAME

```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

**route-map-name (mandatory)**

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

**ipv4-vpn**

Configure IPv4 VPN address family.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn
```

**enabled**

Enable or disable IPv4 VPN Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn
vrouter running ipv4-vpn# enabled true|false
```

**Default value**

true

**unsuppress-map**

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn
vrouter running ipv4-vpn# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

## as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn
vrouter running ipv4-vpn# as-override true|false

```

### Default value

false

## maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn
vrouter running ipv4-vpn# maximum-prefix-out <uint32>

```

## send-community

Send Community attribute to this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn
vrouter running ipv4-vpn# send-community SEND-COMMUNITY

```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

## weight

Set default weight for routes from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn
vrouter running ipv4-vpn# weight <uint16>
```

## allowas-in

Accept as-path with my AS present in it.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn
vrouter running ipv4-vpn# allowas-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

## attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn
vrouter running ipv4-vpn# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

## route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn
vrouter running ipv4-vpn# route-reflector-client true|false
```

**Default value**

false

**route-server-client**

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↪vpn  
vrouter running ipv4-vpn# route-server-client true|false
```

**Default value**

false

**soft-reconfiguration-inbound**

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↪vpn  
vrouter running ipv4-vpn# soft-reconfiguration-inbound true|false
```

**Default value**

false

**update-group-id (state only)**

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-vpn  
↪update-group-id
```

**sub-group-id (state only)**

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-vpn  
↪sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-vpn.  
↳packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-vpn.  
↳accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-vpn.  
↳inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-vpn.  
↳outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳vpn addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn addpath
vrouters running addpath# tx-all-paths true|false

```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn addpath
vrouters running addpath# tx-best-path-per-AS true|false

```

#### Default value

false

### distribute-list

Filter updates to/from this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn
vrouters running ipv4-vpn# distribute-list <distribute-list> access-list ACCESS-LIST

```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

**maximum-prefix**

Maximum number of prefixes to accept from this peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn maximum-prefix
```

**maximum (mandatory)**

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn maximum-prefix
vrouter running maximum-prefix# maximum <uint32>
```

**threshold**

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn maximum-prefix
vrouter running maximum-prefix# threshold <uint8>
```

**Default value**

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳vpn maximum-prefix  
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳vpn maximum-prefix  
vrouter running maximum-prefix# warning-only true|false
```

## Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳vpn nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-  
↳vpn nexthop-self  
vrouter running nexthop-self# force true|false
```

## Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn as-outbound-update
```

### action

Action to apply for ASNs in outbound updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn as-outbound-update
vrouter running as-outbound-update# action ACTION
```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn as-outbound-update
vrouter running as-outbound-update# as-type AS-TYPE
```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

### Default value

private

## filter-list

Establish BGP filters.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn
vrouters running ipv4-vpn# filter-list <filter-list> access-list <as-path-access-list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

## access-list (mandatory)

Access list name.

```
access-list <as-path-access-list>
```

## prefix-list

Filter updates to/from this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↳vpn
vrouters running ipv4-vpn# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

## prefix-list-name (mandatory)

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

## route-map

Apply route map to this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv4-
↪vpn
vrouter running ipv4-vpn# route-map <route-map> route-map-name ROUTE-MAP-NAME

```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

## route-map-name (mandatory)

Route-map name.

```

route-map-name ROUTE-MAP-NAME

```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

## ipv6-unicast

IPv6 unicast address-family associated with the BGP neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↪unicast

```

## enabled

Enable or disable IPv6 unicast Address Family for this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↪unicast
vrouter running ipv6-unicast# enabled true|false

```

## Default value

true

### nexthop-local-unchanged

If true, leave link-local nexthop unchanged for this peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳unicast  
vrouter running ipv6-unicast# nexthop-local-unchanged true|false
```

#### Default value

false

### as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳unicast  
vrouter running ipv6-unicast# as-override true|false
```

#### Default value

false

### maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳unicast  
vrouter running ipv6-unicast# maximum-prefix-out <uint32>
```

### send-community

Send Community attribute to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳unicast  
vrouter running ipv6-unicast# send-community SEND-COMMUNITY
```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

### weight

Set default weight for routes from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast
vrouter running ipv6-unicast# weight <uint16>
```

### allowas-in

Accept as-path with my AS present in it.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast
vrouter running ipv6-unicast# allowas-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

### attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast
vrouter running ipv6-unicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

### route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast
vrouter running ipv6-unicast# route-reflector-client true|false
```

#### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast
vrouter running ipv6-unicast# route-server-client true|false
```

#### Default value

false

### soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast
vrouter running ipv6-unicast# soft-reconfiguration-inbound true|false
```

#### Default value

false

## capability-orf-prefix-list

Advertise prefixlist ORF capability to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast
vrouter running ipv6-unicast# capability-orf-prefix-list CAPABILITY-ORF-PREFIX-LIST
```

| CAPABILITY-ORF-PREFIX-LIST values | Description   |
|-----------------------------------|---|
| both                              | Capability to SEND and RECEIVE the ORF to/from this neighbor. |
| send                              | Capability to SEND the ORF to this neighbor.                  |
| receive                           | Capability to RECEIVE the ORF from this neighbor.             |

## unsuppress-map

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast
vrouter running ipv6-unicast# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

## update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast update-group-id
```

## sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳unicast packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳unicast accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳unicast inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳unicast outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳unicast addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast addpath
vrouters running addpath# tx-all-paths true|false

```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast addpath
vrouters running addpath# tx-best-path-per-AS true|false

```

#### Default value

false

### distribute-list

Filter updates to/from this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast
vrouters running ipv6-unicast# distribute-list <distribute-list> access-list ACCESS-LIST

```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

**maximum-prefix**

Maximum number of prefixes to accept from this peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast maximum-prefix
```

**maximum (mandatory)**

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast maximum-prefix
vrouter running maximum-prefix# maximum <uint32>
```

**threshold**

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast maximum-prefix
vrouter running maximum-prefix# threshold <uint8>
```

**Default value**

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳unicast maximum-prefix  
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳unicast maximum-prefix  
vrouter running maximum-prefix# warning-only true|false
```

### Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳unicast nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳unicast nexthop-self  
vrouter running nexthop-self# force true|false
```

### Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast as-outbound-update
```

### action

Action to apply for ASNs in outbound updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast as-outbound-update
vrouter running as-outbound-update# action ACTION
```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast as-outbound-update
vrouter running as-outbound-update# as-type AS-TYPE
```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

### Default value

private

### filter-list

Establish BGP filters.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast
vrouter running ipv6-unicast# filter-list <filter-list> access-list <as-path-access-
↳list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

### access-list (mandatory)

Access list name.

```
access-list <as-path-access-list>
```

### prefix-list

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast
vrouter running ipv6-unicast# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-
↳NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

### prefix-list-name (mandatory)

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

## route-map

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast
vrouter running ipv6-unicast# route-map <route-map> route-map-name ROUTE-MAP-NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

## route-map-name (mandatory)

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

## default-originate

Originate default route to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast default-originate
```

## route-map

Route-map to specify criteria to originate default.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳unicast default-originate
vrouter running default-originate# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## ipv6-multicast

IPv6 multicast address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast
```

### enabled

Enable or disable IPv6 multicast Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast  
vrouter running ipv6-multicast# enabled true|false
```

### Default value

true

### as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast  
vrouter running ipv6-multicast# as-override true|false
```

### Default value

false

### maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast  
vrouter running ipv6-multicast# maximum-prefix-out <uint32>
```

## send-community

Send Community attribute to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast
vrouter running ipv6-multicast# send-community SEND-COMMUNITY
```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

## weight

Set default weight for routes from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast
vrouter running ipv6-multicast# weight <uint16>
```

## allowas-in

Accept as-path with my AS present in it.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast
vrouter running ipv6-multicast# allowas-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

## attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast
vrouter running ipv6-multicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

## route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast
vrouter running ipv6-multicast# route-reflector-client true|false
```

### Default value

false

## route-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast
vrouter running ipv6-multicast# route-server-client true|false
```

### Default value

false

## soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast
vrouter running ipv6-multicast# soft-reconfiguration-inbound true|false
```

### Default value

false

## capability-orf-prefix-list

Advertise prefixlist ORF capability to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast
vrouter running ipv6-multicast# capability-orf-prefix-list CAPABILITY-ORF-PREFIX-LIST
```

| CAPABILITY-ORF-PREFIX-LIST values | Description   |
|-----------------------------------|---|
| both                              | Capability to SEND and RECEIVE the ORF to/from this neighbor. |
| send                              | Capability to SEND the ORF to this neighbor.                  |
| receive                           | Capability to RECEIVE the ORF from this neighbor.             |

## unsuppress-map

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast
vrouter running ipv6-multicast# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast addpath  
vrouter running addpath# tx-all-paths true|false
```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast addpath  
vrouter running addpath# tx-best-path-per-AS true|false
```

#### Default value

false

## distribute-list

Filter updates to/from this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast
vrouters running ipv6-multicast# distribute-list <distribute-list> access-list ACCESS-
↳LIST

```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

## access-list (mandatory)

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

## maximum-prefix

Maximum number of prefixes to accept from this peer.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast maximum-prefix

```

## maximum (mandatory)

Maximum number of prefix limit.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast maximum-prefix
vrouters running maximum-prefix# maximum <uint32>

```

## threshold

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast maximum-prefix  
vrouter running maximum-prefix# threshold <uint8>
```

### Default value

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast maximum-prefix  
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast maximum-prefix  
vrouter running maximum-prefix# warning-only true|false
```

### Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast nexthop-self
vrouters running nexthop-self# force true|false
```

### Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast as-outbound-update
```

## action

Action to apply for ASNs in outbound updates.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast as-outbound-update
vrouters running as-outbound-update# action ACTION
```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast as-outbound-update
vrouters running as-outbound-update# as-type AS-TYPE
```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

**Default value**

private

**filter-list**

Establish BGP filters.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast
vrouters running ipv6-multicast# filter-list <filter-list> access-list <as-path-access-
↳list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list <as-path-access-list>
```

**prefix-list**

Filter updates to/from this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast
vrouters running ipv6-multicast# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-
↳NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**prefix-list-name (mandatory)**

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

**route-map**

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast
vrouter running ipv6-multicast# route-map <route-map> route-map-name ROUTE-MAP-NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

**route-map-name (mandatory)**

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

**default-originate**

Originate default route to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳multicast default-originate
```

## route-map

Route-map to specify criteria to originate default.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳multicast default-originate  
vrouter running default-originate# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## ipv6-labeled-unicast

IPv6 labeled unicast address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast
```

## enabled

Enable or disable IPv6 labeled unicast Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast  
vrouter running ipv6-labeled-unicast# enabled true|false
```

### Default value

true

## as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast  
vrouter running ipv6-labeled-unicast# as-override true|false
```

### Default value

false

## maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast
vrouters running ipv6-labeled-unicast# maximum-prefix-out <uint32>
```

## send-community

Send Community attribute to this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast
vrouters running ipv6-labeled-unicast# send-community SEND-COMMUNITY
```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

## weight

Set default weight for routes from this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast
vrouters running ipv6-labeled-unicast# weight <uint16>
```

## allowas-in

Accept as-path with my AS present in it.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast
vrouters running ipv6-labeled-unicast# allowas-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

### attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast
vrouters running ipv6-labeled-unicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

### route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast
vrouters running ipv6-labeled-unicast# route-reflector-client true|false
```

#### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast
vrouters running ipv6-labeled-unicast# route-server-client true|false
```

#### Default value

false

## soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast
vrouter running ipv6-labeled-unicast# soft-reconfiguration-inbound true|false
```

### Default value

false

## capability-orf-prefix-list

Advertise prefixlist ORF capability to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast
vrouter running ipv6-labeled-unicast# capability-orf-prefix-list CAPABILITY-ORF-PREFIX-
↳LIST
```

| CAPABILITY-ORF-PREFIX-LIST values | Description   |
|-----------------------------------|---|
| both                              | Capability to SEND and RECEIVE the ORF to/from this neighbor. |
| send                              | Capability to SEND the ORF to this neighbor.                  |
| receive                           | Capability to RECEIVE the ORF from this neighbor.             |

## unsuppress-map

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast
vrouter running ipv6-labeled-unicast# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast addpath  
vrouter running addpath# tx-all-paths true|false
```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast addpath  
vrouter running addpath# tx-best-path-per-AS true|false
```

#### Default value

false

### distributed-list

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast
vrouter running ipv6-labeled-unicast# distribute-list <distribute-list> access-list_
↳ACCESS-LIST
```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

### access-list (mandatory)

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

### maximum-prefix

Maximum number of prefixes to accept from this peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast maximum-prefix
```

### maximum (mandatory)

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast maximum-prefix
vrouter running maximum-prefix# maximum <uint32>
```

## threshold

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast maximum-prefix  
vrouter running maximum-prefix# threshold <uint8>
```

### Default value

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast maximum-prefix  
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast maximum-prefix  
vrouter running maximum-prefix# warning-only true|false
```

### Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast nexthop-self
vrouter running nexthop-self# force true|false

```

### Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast as-outbound-update

```

## action

Action to apply for ASNs in outbound updates.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast as-outbound-update
vrouter running as-outbound-update# action ACTION

```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast as-outbound-update
vrouter running as-outbound-update# as-type AS-TYPE

```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

**Default value**

private

**filter-list**

Establish BGP filters.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast
vrouter running ipv6-labeled-unicast# filter-list <filter-list> access-list <as-path-
↳access-list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list <as-path-access-list>
```

**prefix-list**

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast
vrouter running ipv6-labeled-unicast# prefix-list <prefix-list> prefix-list-name.
↳PREFIX-LIST-NAME
```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

### prefix-list-name (mandatory)

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

### route-map

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast
vrouter running ipv6-labeled-unicast# route-map <route-map> route-map-name ROUTE-MAP-
↳NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

### route-map-name (mandatory)

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

### default-originate

Originate default route to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳labeled-unicast default-originate
```

## route-map

Route-map to specify criteria to originate default.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳labeled-unicast default-originate  
vrouter running default-originate# route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## ipv6-flowspec

IPv6 Flowspec address-family associated with the BGP neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳flowspec
```

## enabled

Enable or disable IPv6 Flowspec Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳flowspec  
vrouter running ipv6-flowspec# enabled true|false
```

### Default value

true

## route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳flowspec  
vrouter running ipv6-flowspec# route-reflector-client true|false
```

### Default value

false

### route-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳flowspec  
vrouter running ipv6-flowspec# route-server-client true|false
```

#### Default value

false

### soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳flowspec  
vrouter running ipv6-flowspec# soft-reconfiguration-inbound true|false
```

#### Default value

false

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳flowspec update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳flowspec sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳flowspec packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳flowspec accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳flowspec inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳flowspec outbound-ebgp-requires-policy
```

### filter-list

Establish BGP filters.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳flowspec  
vrouter running ipv6-flowspec# filter-list <filter-list> access-list <as-path-access-  
↳list...
```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

## access-list (mandatory)

Access list name.

```
access-list <as-path-access-list>
```

## prefix-list

Filter updates to/from this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳flowspec
vrouters running ipv6-flowspec# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-
↳NAME

```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

## prefix-list-name (mandatory)

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

## route-map

Apply route map to this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳flowspec
vrouters running ipv6-flowspec# route-map <route-map> route-map-name ROUTE-MAP-NAME

```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

**route-map-name (mandatory)**

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

**ipv6-vpn**

Configure IPv6 VPN address family.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn
```

**enabled**

Enable or disable IPv6 VPN Address Family for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn
vrouter running ipv6-vpn# enabled true|false
```

**Default value**

true

**unsuppress-map**

Route-map to selectively unsuppress suppressed routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn
vrouter running ipv6-vpn# unsuppress-map UNSUPPRESS-MAP
```

|                |                 |
|----------------|-----------------|
| UNSUPPRESS-MAP | Route map name. |
|----------------|-----------------|

## as-override

If true, disable checking if nexthop is connected on ebgp sessions.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn
vrouter running ipv6-vpn# as-override true|false

```

### Default value

false

## maximum-prefix-out

Sets a maximum number of prefixes we can send to a given peer.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn
vrouter running ipv6-vpn# maximum-prefix-out <uint32>

```

## send-community

Send Community attribute to this neighbor.

```

vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn
vrouter running ipv6-vpn# send-community SEND-COMMUNITY

```

| SEND-COMMUNITY values | Description  |
|-----------------------|--|
| all                   | Send Standard and Extended Community attributes.       |
| extended              | Send Extended Community attributes.                    |
| standard              | Send Standard Community attributes.                    |
| none                  | Do not send Standard or Extended Community attributes. |

### Default value

all

## weight

Set default weight for routes from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn
vrouter running ipv6-vpn# weight <uint16>
```

## allowas-in

Accept as-path with my AS present in it.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn
vrouter running ipv6-vpn# allowas-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

## attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn
vrouter running ipv6-vpn# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

## route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn
vrouter running ipv6-vpn# route-reflector-client true|false
```

**Default value**

false

**route-server-client**

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↪vpn  
vrouter running ipv6-vpn# route-server-client true|false
```

**Default value**

false

**soft-reconfiguration-inbound**

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↪vpn  
vrouter running ipv6-vpn# soft-reconfiguration-inbound true|false
```

**Default value**

false

**update-group-id (state only)**

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-vpn  
↪update-group-id
```

**sub-group-id (state only)**

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-vpn  
↪sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-vpn.  
↳packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-vpn.  
↳accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-vpn.  
↳inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-vpn.  
↳outbound-ebgp-requires-policy
```

### addpath

Configure addpath.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳vpn addpath
```

### tx-all-paths

If true, use addpath to advertise all paths to a neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn addpath
vrouters running addpath# tx-all-paths true|false

```

#### Default value

false

### tx-best-path-per-AS

If true, use addpath to advertise the bestpath per each neighboring AS.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn addpath
vrouters running addpath# tx-best-path-per-AS true|false

```

#### Default value

false

### distribute-list

Filter updates to/from this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn
vrouters running ipv6-vpn# distribute-list <distribute-list> access-list ACCESS-LIST

```

| <distribute-list> values | Description              |
|--------------------------|--------------------------|
| in                       | Filter incoming updates. |
| out                      | Filter outgoing updates. |

**access-list (mandatory)**

Access list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

**maximum-prefix**

Maximum number of prefixes to accept from this peer.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn maximum-prefix
```

**maximum (mandatory)**

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn maximum-prefix
vrouter running maximum-prefix# maximum <uint32>
```

**threshold**

Threshold value (%) at which to generate a warning msg.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn maximum-prefix
vrouter running maximum-prefix# threshold <uint8>
```

**Default value**

75

## restart

Restart interval in minutes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳vpn maximum-prefix  
vrouter running maximum-prefix# restart <uint16>
```

## warning-only

If true, only give warning message when limit is exceeded.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳vpn maximum-prefix  
vrouter running maximum-prefix# warning-only true|false
```

## Default value

false

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳vpn nexthop-self
```

## force

If true, set the next hop to self for reflected routes.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-  
↳vpn nexthop-self  
vrouter running nexthop-self# force true|false
```

## Default value

false

## as-outbound-update

Remove or replace ASNs in outbound updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn as-outbound-update
```

### action

Action to apply for ASNs in outbound updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn as-outbound-update
vrouter running as-outbound-update# action ACTION
```

| ACTION values | Description                                  |
|---------------|--|
| replace       | Replace ASNs in outbound updates by our ASN. |
| remove        | Remove ASN in outbound updates.              |

### Default value

remove

## as-type

Apply to private AS numbers or all.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn as-outbound-update
vrouter running as-outbound-update# as-type AS-TYPE
```

| AS-TYPE values | Description                       |
|----------------|-----------------------------------|
| private        | Apply to private AS numbers only. |
| all            | Apply to all AS numbers.          |

### Default value

private

## filter-list

Establish BGP filters.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn
vrouters running ipv6-vpn# filter-list <filter-list> access-list <as-path-access-list...

```

| <filter-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

## access-list (mandatory)

Access list name.

```
access-list <as-path-access-list>
```

## prefix-list

Filter updates to/from this neighbor.

```

vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn
vrouters running ipv6-vpn# prefix-list <prefix-list> prefix-list-name PREFIX-LIST-NAME

```

| <prefix-list> values | Description              |
|----------------------|--------------------------|
| in                   | Filter incoming updates. |
| out                  | Filter outgoing updates. |

## prefix-list-name (mandatory)

Name of the prefix list.

```
prefix-list-name PREFIX-LIST-NAME
```

|                  |                   |
|------------------|-------------------|
| PREFIX-LIST-NAME | Prefix list name. |
|------------------|-------------------|

## route-map

Apply route map to this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family ipv6-
↳vpn
vrouters running ipv6-vpn# route-map <route-map> route-map-name ROUTE-MAP-NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

## route-map-name (mandatory)

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

## l2vpn-evpn

Configure L2VPN EVPN address family.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-
↳evpn
```

## enabled

Enable or disable L2VPN EVPN Address Family for this neighbor.

```
vrouters running config# vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-
↳evpn
vrouters running l2vpn-evpn# enabled true|false
```

## Default value

true

## nexthop-self

Disable the next hop calculation for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-  
↳evpn  
vrouter running l2vpn-evpn# nexthop-self true|false
```

### Default value

false

## route-reflector-client

If true, configure a neighbor as Route Reflector client. This only applies to internal neighbors (IGP).

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-  
↳evpn  
vrouter running l2vpn-evpn# route-reflector-client true|false
```

### Default value

false

## route-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-  
↳evpn  
vrouter running l2vpn-evpn# route-server-client true|false
```

### Default value

false

## soft-reconfiguration-inbound

If true, allow inbound soft reconfiguration for this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-  
↳evpn  
vrouter running l2vpn-evpn# soft-reconfiguration-inbound true|false
```

### Default value

false

### allows-in

Accept as-path with my AS present in it.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-
↳evpn
vrouter running l2vpn-evpn# allows-in ALLOWAS-IN
```

| ALLOWAS-IN values | Description  |
|-------------------|--|
| <uint8>           | No description.  |
| origin            | Only accept my AS in the as-path if the route was originated in my AS. |

### attribute-unchanged

BGP attribute is propagated unchanged to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-
↳evpn
vrouter running l2vpn-evpn# attribute-unchanged ATTRIBUTE-UNCHANGED
```

| ATTRIBUTE-UNCHANGED values | Description        |
|----------------------------|--------------------|
| as-path                    | As-path attribute. |
| med                        | Med attribute.     |
| nexthop                    | Nexthop attribute. |

### update-group-id (state only)

Update group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-
↳evpn update-group-id
```

### sub-group-id (state only)

Sub-group identifier.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-  
↳evpn sub-group-id
```

### packet-queue-length (state only)

Packet queue length.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-  
↳evpn packet-queue-length
```

### accepted-prefix (state only)

Accepted prefix counter.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-  
↳evpn accepted-prefix
```

### inbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that incoming BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-  
↳evpn inbound-ebgp-requires-policy
```

### outbound-ebgp-requires-policy (state only)

The presence of this comment informs the user that outgoing BGP updates are discarded, as per RFC8212 behaviour.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-  
↳evpn outbound-ebgp-requires-policy
```

### route-map

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family l2vpn-
↳ evpn
vrouter running l2vpn-evpn# route-map <route-map> route-map-name ROUTE-MAP-NAME
```

| <route-map> values | Description                   |
|--------------------|-------------------------------|
| in                 | Apply map to incoming routes. |
| out                | Apply map to outbound routes. |

### route-map-name (mandatory)

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

### connections (state only)

Established/dropped connections statistics.

#### established (state only)

Number of established connections.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> connections established
```

#### dropped (state only)

Number of dropped connections.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> connections dropped
```

**local-host (state only)**

Local host data.

**name (state only)**

Local host name.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> local-host name
```

**port (state only)**

Local host port.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> local-host port
```

**remote-host (state only)**

Remote host data.

**name (state only)**

Remote host name.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> remote-host name
```

**port (state only)**

Remote host port.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> remote-host port
```

**nexthop (state only)**

Nexthop data.

**address (state only)**

Nexthop IP address.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> nexthop address
```

**global-address (state only)**

Nexthop global IP address.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> nexthop global-address
```

**local-address (state only)**

Nexthop local IP address.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> nexthop local-address
```

**thread (state only)**

Read/Write thread.

**read-enabled (state only)**

Read thread status.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> thread read-enabled
```

### write-enabled (state only)

Write thread status.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> thread write-enabled
```

### message-statistics (state only)

Neighbor messages statistics.

### packet-wait-process (state only)

Number of packets waiting to be processed.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics_
↳packet-wait-process
```

### packet-wait-written (state only)

Number of packets waiting to be written.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics_
↳packet-wait-written
```

### open-sent (state only)

BGP open messages sent.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics open-
↳sent
```

### opens-received (state only)

BGP open messages received.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics opens-
↳received
```

### notifications-sent (state only)

Notifications messages sent.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics_
↳notifications-sent
```

### notifications-received (state only)

Notification messages received.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics_
↳notifications-received
```

### updates-sent (state only)

Update messages sent.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics_
↳updates-sent
```

### updates-received (state only)

Update messages received.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics_
↳updates-received
```

### keepalives-sent (state only)

Keepalive messages sent.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics_
↳keepalives-sent
```

### keepalives-received (state only)

Keepalive messages received.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics_
↳keepalives-received
```

### route-refresh-sent (state only)

Route refresh messages sent.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics route-
↳refresh-sent
```

### route-refresh-received (state only)

Route refresh messages received.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics route-
↳refresh-received
```

### capability-sent (state only)

Capability messages sent.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics_
↳capability-sent
```

### capability-received (state only)

Capability messages received.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics_
↳capability-received
```

### total-sent (state only)

Total messages sent.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics total-  
↪sent
```

### total-received (state only)

Total messages received.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> message-statistics total-  
↪received
```

### rpki

BGP RPKI configuration.

```
vrouter running config# vrf <vrf> routing bgp rpki
```

### enabled

Enable/Disable BGP RPKI configuration.

```
vrouter running config# vrf <vrf> routing bgp rpki  
vrouter running rpki# enabled true|false
```

### Default value

true

### expire-interval

Set expire interval in seconds.

```
vrouter running config# vrf <vrf> routing bgp rpki  
vrouter running rpki# expire-interval <uint32>
```

### Default value

7200

## polling-period

Set polling period in seconds.

```
vrouter running config# vrf <vrf> routing bgp rpki  
vrouter running rpki# polling-period <uint32>
```

### Default value

3600

## retry-interval

Set retry interval in seconds.

```
vrouter running config# vrf <vrf> routing bgp rpki  
vrouter running rpki# retry-interval <uint16>
```

### Default value

600

## cache-server

Configure a cache server.

```
vrouter running config# vrf <vrf> routing bgp rpki  
vrouter running rpki# cache-server <uint8> address ADDRESS \  
...  ssh port <uint16> user-name <string> key <string> \  
...  tcp port <uint16>
```

|         |                                 |
|---------|---------------------------------|
| <uint8> | Preference of the cache server. |
|---------|---------------------------------|

## address (mandatory)

IP address or hostname of the cache server.

```
address ADDRESS
```

| ADDRESS values | Description  |
|----------------|--|
| <A.B.C.D>      | IPv4 address.  |
| <fqdn>         | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |

### status (state only)

Cache server status.

```
vrouter> show state vrf <vrf> routing bgp rpki cache-server <uint8> status
```

### ssh

Use SSH protocol for this cache server.

```
ssh port <uint16> user-name <string> key <string>
```

### port (mandatory)

Port number.

```
port <uint16>
```

**user-name (mandatory)**

SSH user name.

```
user-name <string>
```

**key (mandatory)**

SSH key pair name.

```
key <string>
```

**tcp**

Use TCP protocol for this cache server.

```
tcp port <uint16>
```

**port (mandatory)**

Port number.

```
port <uint16>
```

**ldp**

---

**Note:** requires a Turbo Router Network License.

---

LDP configuration.

```
vrouter running config# vrf <vrf> routing mpls ldp
```

## enabled

Enable or disable LDP.

```
vrouter running config# vrf <vrf> routing mpls ldp
vrouter running ldp# enabled true|false
```

### Default value

true

## router-id

LSR Id in IPv4 address format.

```
vrouter running config# vrf <vrf> routing mpls ldp
vrouter running ldp# router-id ROUTER-ID
```

|           |                  |
|-----------|------------------|
| ROUTER-ID | An IPv4 address. |
|-----------|------------------|

## discovery

Discovery parameters.

```
vrouter running config# vrf <vrf> routing mpls ldp discovery
```

## hello

LDP Link Hellos.

```
vrouter running config# vrf <vrf> routing mpls ldp discovery hello
```

## holdtime

Hello holdtime in seconds.

```
vrouter running config# vrf <vrf> routing mpls ldp discovery hello
vrouter running hello# holdtime <uint16>
```

### Default value

15

## interval

Hello interval in seconds.

```
vrouter running config# vrf <vrf> routing mpls ldp discovery hello
vrouter running hello# interval <uint16>
```

### Default value

5

## dual-stack

Configure dual stack parameters.

```
vrouter running config# vrf <vrf> routing mpls ldp dual-stack
```

## cisco-interop

Use Cisco non-compliant format to send and interpret the Dual-Stack capability TLV.

```
vrouter running config# vrf <vrf> routing mpls ldp dual-stack
vrouter running dual-stack# cisco-interop true|false
```

### Default value

false

## transport-preference

Configure preferred address family for TCP transport connection with neighbor.

```
vrouter running config# vrf <vrf> routing mpls ldp dual-stack
vrouter running dual-stack# transport-preference TRANSPORT-PREFERENCE
```

| TRANSPORT-PREFERENCE values | Description |
|-----------------------------|-------------|
| ipv4                        | IPv4.       |
| ipv6                        | IPv6.       |

### Default value

ipv6

## neighbor

Configure neighbor parameters.

```
vrouter running config# vrf <vrf> routing mpls ldp neighbor <neighbor>
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv4 address. |
|------------|------------------|

## password

The password.

```
vrouter running config# vrf <vrf> routing mpls ldp neighbor <neighbor>  
vrouter running neighbor <neighbor># password <string>
```

## ttl-security

LDP ttl security check.

```
vrouter running config# vrf <vrf> routing mpls ldp neighbor <neighbor> ttl-security
```

## hops

Maximum number of IP hops.

```
vrouter running config# vrf <vrf> routing mpls ldp neighbor <neighbor> ttl-security  
vrouter running ttl-security# hops <uint8>
```

## disable

Disable ttl security.

```
vrouter running config# vrf <vrf> routing mpls ldp neighbor <neighbor> ttl-security  
vrouter running ttl-security# disable true|false
```

## Default value

false

## address-family

Configure Address Family and its parameters.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family
```

## ipv4

IPv4.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4
```

## session-holdtime

Session holdtime in seconds.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4  
vrouter running ipv4# session-holdtime <uint16>
```

### Default value

180

## discovery

Discovery parameters.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 discovery
```

## transport-address (mandatory)

Transport address for TCP connection.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 discovery  
vrouter running discovery# transport-address TRANSPORT-ADDRESS
```

|                   |                  |
|-------------------|------------------|
| TRANSPORT-ADDRESS | An IPv4 address. |
|-------------------|------------------|

## hello

LDP Link Hellos.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 discovery hello
```

## holdtime

Hello holdtime in seconds.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 discovery hello  
vrouter running hello# holdtime <uint16>
```

## Default value

15

## interval

Hello interval in seconds.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 discovery hello  
vrouter running hello# interval <uint16>
```

## Default value

5

## interface

Enable LDP on an interface.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 interface  
↪<interface>
```

|             |                    |
|-------------|--------------------|
| <interface> | An interface name. |
|-------------|--------------------|

## label

Configure label control and policies.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 label
```

## local

Local label control and policies.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 label local
```

## advertise

Configure outbound label advertisement control.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 label local
↳advertise
```

## for

IP access-list for destination prefixes.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 label local
↳advertise
vrouter running advertise# for FOR
```

|     |                   |
|-----|-------------------|
| FOR | Access list name. |
|-----|-------------------|

## to

IP Access-list specifying controls on LDP Peers.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 label local
↳advertise
vrouter running advertise# to TO
```

|    |                   |
|----|-------------------|
| TO | Access list name. |
|----|-------------------|

## explicit-null

Configure explicit-null advertisement.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 label local_
↳advertise explicit-null
```

### for

IP access-list for destination prefixes.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 label local_
↳advertise explicit-null
vrouter running explicit-null# for FOR
```

|     |                   |
|-----|-------------------|
| FOR | Access list name. |
|-----|-------------------|

## allocate

Configure label allocation control.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 label local_
↳allocate
```

### for

IP access-list.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 label local_
↳allocate
vrouter running allocate# for FOR
```

|     |                   |
|-----|-------------------|
| FOR | Access list name. |
|-----|-------------------|

## host-routes

Allocate local label for host routes only.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 label local_
↳allocate
vrouter running allocate# host-routes
```

## remote

Remote/peer label control and policies.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 label remote
```

## accept

Configure inbound label acceptance control.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 label remote_
↳accept
```

## for

IP access-list for destination prefixes.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 label remote_
↳accept
vrouter running accept# for FOR
```

|     |                   |
|-----|-------------------|
| FOR | Access list name. |
|-----|-------------------|

## from

Neighbor from whom to accept label advertisement.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv4 label remote_
↳accept
vrouter running accept# from FROM
```

|      |                   |
|------|-------------------|
| FROM | Access list name. |
|------|-------------------|

## ipv6

IPv6.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6
```

## session-holdtime

Session holdtime in seconds.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6  
vrouter running ipv6# session-holdtime <uint16>
```

## Default value

180

## discovery

Discovery parameters.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 discovery
```

## transport-address (mandatory)

Transport address for TCP connection.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 discovery  
vrouter running discovery# transport-address TRANSPORT-ADDRESS
```

|                   |                  |
|-------------------|------------------|
| TRANSPORT-ADDRESS | An IPv6 address. |
|-------------------|------------------|

## hello

LDP Link Hellos.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 discovery hello
```

## holdtime

Hello holdtime in seconds.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 discovery hello  
vrouter running hello# holdtime <uint16>
```

### Default value

15

## interval

Hello interval in seconds.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 discovery hello  
vrouter running hello# interval <uint16>
```

### Default value

5

## interface

Enable LDP on an interface.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 interface  
↪<interface>
```

|             |                    |
|-------------|--------------------|
| <interface> | An interface name. |
|-------------|--------------------|

## label

Configure label control and policies.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 label
```

## local

Local label control and policies.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 label local
```

## advertise

Configure outbound label advertisement control.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 label local  
↳advertise
```

## for

IP access-list for destination prefixes.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 label local  
↳advertise  
vrouter running advertise# for FOR
```

|     |                   |
|-----|-------------------|
| FOR | Access list name. |
|-----|-------------------|

## to

IP Access-list specifying controls on LDP Peers.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 label local  
↳advertise  
vrouter running advertise# to TO
```

|    |                   |
|----|-------------------|
| TO | Access list name. |
|----|-------------------|

## explicit-null

Configure explicit-null advertisement.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 label local_
↳advertise explicit-null
```

### for

IP access-list for destination prefixes.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 label local_
↳advertise explicit-null
vrouter running explicit-null# for FOR
```

|     |                   |
|-----|-------------------|
| FOR | Access list name. |
|-----|-------------------|

## allocate

Configure label allocation control.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 label local_
↳allocate
```

### for

IP access-list.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 label local_
↳allocate
vrouter running allocate# for FOR
```

|     |                   |
|-----|-------------------|
| FOR | Access list name. |
|-----|-------------------|

## host-routes

Allocate local label for host routes only.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 label local_
↳allocate
vrouter running allocate# host-routes
```

## remote

Remote/peer label control and policies.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 label remote
```

## accept

Configure inbound label acceptance control.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 label remote_
↳accept
```

## for

IP access-list for destination prefixes.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 label remote_
↳accept
vrouter running accept# for FOR
```

|     |                   |
|-----|-------------------|
| FOR | Access list name. |
|-----|-------------------|

## from

Neighbor from whom to accept label advertisement.

```
vrouter running config# vrf <vrf> routing mpls ldp address-family ipv6 label remote_
↳accept
vrouter running accept# from FROM
```

|      |                   |
|------|-------------------|
| FROM | Access list name. |
|------|-------------------|

## nhrp

---

**Note:** requires a Turbo Router Network License.

---

NHRP configuration.

```
vrouter running config# vrf <vrf> routing nhrp
```

### enabled

Enable or disable NHRP.

```
vrouter running config# vrf <vrf> routing nhrp
vrouter running nhrp# enabled true|false
```

### Default value

true

### hub-mode

Enable or disable the NHRP hub redirect capacity.

```
vrouter running config# vrf <vrf> routing nhrp
vrouter running nhrp# hub-mode true|false
```

### Default value

false

### nhrp4-cache (state only)

Nexthop Cache Entry.

**type (state only)**

NHRP cache type.

```
vrouter> show state vrf <vrf> routing nhrp nhrp4-cache <string> <protocol> type
```

**nbma (state only)**

NBMA IP address.

```
vrouter> show state vrf <vrf> routing nhrp nhrp4-cache <string> <protocol> nbma
```

**timeout (state only)**

Tell if cache entry has timeout.

```
vrouter> show state vrf <vrf> routing nhrp nhrp4-cache <string> <protocol> timeout
```

**auth (state only)**

Tell if cache entry has authentication procedure.

```
vrouter> show state vrf <vrf> routing nhrp nhrp4-cache <string> <protocol> auth
```

**used (state only)**

Tell if cache entry is used.

```
vrouter> show state vrf <vrf> routing nhrp nhrp4-cache <string> <protocol> used
```

**identity (state only)**

Identity of the connection of the cache entry.

```
vrouter> show state vrf <vrf> routing nhrp nhrp4-cache <string> <protocol> identity
```

### **nhrp6-cache (state only)**

Nexthop Cache Entry.

#### **type (state only)**

NHRP cache type.

```
vrouter> show state vrf <vrf> routing nhrp nhrp6-cache <string> <protocol> type
```

#### **nbma (state only)**

NBMA IP address.

```
vrouter> show state vrf <vrf> routing nhrp nhrp6-cache <string> <protocol> nbma
```

#### **timeout (state only)**

Tell if cache entry has timeout.

```
vrouter> show state vrf <vrf> routing nhrp nhrp6-cache <string> <protocol> timeout
```

#### **auth (state only)**

Tell if cache entry has authentication procedure.

```
vrouter> show state vrf <vrf> routing nhrp nhrp6-cache <string> <protocol> auth
```

#### **used (state only)**

Tell if cache entry is used.

```
vrouter> show state vrf <vrf> routing nhrp nhrp6-cache <string> <protocol> used
```

### identity (state only)

Identity of the connection of the cache entry.

```
vrouter> show state vrf <vrf> routing nhrp nhrp6-cache <string> <protocol> identity
```

### nhrp4-nhs (state only)

Nexthop Cache Entry.

### fqdn (state only)

Fully Qualified Domain Name of Server.

```
vrouter> show state vrf <vrf> routing nhrp nhrp4-nhs <string> <nbma> fqdn
```

### protocol (state only)

Protocol IPv4 Address.

```
vrouter> show state vrf <vrf> routing nhrp nhrp4-nhs <string> <nbma> protocol
```

### nhrp6-nhs (state only)

Nexthop Cache Entry.

### fqdn (state only)

Fully Qualified Domain Name of Server.

```
vrouter> show state vrf <vrf> routing nhrp nhrp6-nhs <string> <nbma> fqdn
```

**protocol (state only)**

Protocol IPv6 Address.

```
vrouter> show state vrf <vrf> routing nhrp nhrp6-nhs <string> <nbma> protocol
```

**connection (state only)**

NHRP Connection.

**notifier-active (state only)**

Whether the connection is active.

```
vrouter> show state vrf <vrf> routing nhrp connection <string> <string> notifier-active
```

**sa-count (state only)**

Number of child SAs for this connection.

```
vrouter> show state vrf <vrf> routing nhrp connection <string> <string> sa-count
```

**identity (state only)**

IKE remote identity of the connection.

```
vrouter> show state vrf <vrf> routing nhrp connection <string> <string> identity
```

**ospf**

---

**Note:** requires a Turbo Router Network License.

---

OSPF configuration.

```
vrouter running config# vrf <vrf> routing ospf
```

### enabled

Enable or disable OSPF.

```
vrouter running config# vrf <vrf> routing ospf
vrouter running ospf# enabled true|false
```

#### Default value

true

### router-id

OSPF router-id in IP address format.

```
vrouter running config# vrf <vrf> routing ospf
vrouter running ospf# router-id ROUTER-ID
```

|           |                  |
|-----------|------------------|
| ROUTER-ID | An IPv4 address. |
|-----------|------------------|

### abr-type

OSPF ABR type.

```
vrouter running config# vrf <vrf> routing ospf
vrouter running ospf# abr-type ABR-TYPE
```

| ABR-TYPE values | Description                            |
|-----------------|--|
| cisco           | Alternative ABR, cisco implementation. |
| ibm             | Alternative ABR, IBM implementation.   |
| shortcut        | Shortcut ABR.                          |
| standard        | Standard behavior (RFC2328).           |

#### Default value

cisco

### write-multiplier

Maximum number of interface serviced per write.

```
vrouter running config# vrf <vrf> routing ospf  
vrouter running ospf# write-multiplier <uint8>
```

#### Default value

20

### auto-cost

Calculate OSPF interface cost according to reference bandwidth (Mbits per second).

```
vrouter running config# vrf <vrf> routing ospf  
vrouter running ospf# auto-cost <uint32>
```

#### Default value

100000

### opaque-lsa

Enable or disable opaque LSA capability.

```
vrouter running config# vrf <vrf> routing ospf  
vrouter running ospf# opaque-lsa true|false
```

### compatible-rfc1583

Enable or disable compatibility with RFC 1583.

```
vrouter running config# vrf <vrf> routing ospf  
vrouter running ospf# compatible-rfc1583 true|false
```

## default-metric

Set metric of redistributed routes.

```
vrouter running config# vrf <vrf> routing ospf
vrouter running ospf# default-metric <uint32>
```

## log-adjacency-changes

Log changes in adjacency state.

```
vrouter running config# vrf <vrf> routing ospf
vrouter running ospf# log-adjacency-changes LOG-ADJACENCY-CHANGES
```

| LOG-ADJACENCY-CHANGES values | Description            |
|------------------------------|------------------------|
| standard                     | Standard logs.         |
| detail                       | Log all state changes. |

## refresh-timer

LSA refresh interval (in seconds).

```
vrouter running config# vrf <vrf> routing ospf
vrouter running ospf# refresh-timer <uint16>
```

### Default value

10

## area

OSPF area parameters.

```
vrouter running config# vrf <vrf> routing ospf area <area>
```

| <area> values | Description      |
|---------------|------------------|
| <uint32>      | OSPF area ID.    |
| <A.B.C.D>     | An IPv4 address. |

## default-cost

Default summary cost of a NSSA or stub area.

```
vrouter running config# vrf <vrf> routing ospf area <area>
vrouter running area <area># default-cost <uint32>
```

## export-list

Set the filter for networks announced to other areas (access-list name).

```
vrouter running config# vrf <vrf> routing ospf area <area>
vrouter running area <area># export-list <string>
```

## import-list

Set the filter for networks from other areas announced to the specified one (access-list name).

```
vrouter running config# vrf <vrf> routing ospf area <area>
vrouter running area <area># import-list <string>
```

## shortcut

Enable/Disable shortcutting through the area.

```
vrouter running config# vrf <vrf> routing ospf area <area>
vrouter running area <area># shortcut true|false
```

## nssa

Configure OSPF area as nssa.

```
vrouter running config# vrf <vrf> routing ospf area <area>
vrouter running area <area># nssa summary true|false translate TRANSLATE
```

**summary**

Inject inter-area routes into nssa.

```
summary true|false
```

**Default value**

true

**translate**

NSSA-ABR translate.

```
translate TRANSLATE
```

| TRANSLATE values | Description  |
|------------------|--|
| always           | Configure NSSA-ABR to always translate.              |
| candidate        | Configure NSSA-ABR for translate election (default). |
| never            | Configure NSSA-ABR to never translate.               |

**Default value**

candidate

**stub**

Configure OSPF area as stub.

```
vrouter running config# vrf <vrf> routing ospf area <area>
vrouter running area <area># stub summary true|false
```

**summary**

Inject inter-area routes into stub.

```
summary true|false
```

**Default value**

true

## virtual-link

Virtual links.

```
vrouter running config# vrf <vrf> routing ospf area <area> virtual-link <virtual-link>
```

|                |                  |
|----------------|------------------|
| <virtual-link> | An IPv4 address. |
|----------------|------------------|

## authentication

Enable authentication.

```
vrouter running config# vrf <vrf> routing ospf area <area> authentication
```

## message-digest

If true, use message-digest authentication.

```
vrouter running config# vrf <vrf> routing ospf area <area> authentication  
vrouter running authentication# message-digest true|false
```

## filter-list

Filter networks between OSPF areas.

```
vrouter running config# vrf <vrf> routing ospf area <area> filter-list
```

## input

Filter networks sent to this area (prefix-list name).

```
vrouter running config# vrf <vrf> routing ospf area <area> filter-list  
vrouter running filter-list# input <string>
```

## output

Filter networks sent from this area (prefix-list name).

```
vrouter running config# vrf <vrf> routing ospf area <area> filter-list
vrouter running filter-list# output <string>
```

## range

Summarize routes matching address/mask (border routers only).

```
vrouter running config# vrf <vrf> routing ospf area <area>
vrouter running area <area># range <range> action ACTION cost <uint32>
```

|         |  |
|---------|--|
| <range> | An IPv4 prefix: address and CIDR mask. |
|---------|--|

## action

Advertise this range, do not advertise or announce as another prefix.

```
action ACTION
```

| ACTION values | Description                            |
|---------------|--|
| advertise     | Advertise this range (default).        |
| not-advertise | Do not advertise this range.           |
| <A.B.C.D/M>   | An IPv4 prefix: address and CIDR mask. |

## Default value

advertise

## cost

User specified metric for this range.

```
cost <uint32>
```

## default-information

Control distribution of default information.

```
vrouter running config# vrf <vrf> routing ospf default-information
```

## always

If true, always advertise default route.

```
vrouter running config# vrf <vrf> routing ospf default-information  
vrouter running default-information# always true|false
```

## metric

OSPF default metric.

```
vrouter running config# vrf <vrf> routing ospf default-information  
vrouter running default-information# metric <uint32>
```

## metric-type

OSPF metric type for default routes.

```
vrouter running config# vrf <vrf> routing ospf default-information  
vrouter running default-information# metric-type <uint8>
```

## Default value

2

## route-map

Route map reference.

```
vrouter running config# vrf <vrf> routing ospf default-information  
vrouter running default-information# route-map <string>
```

## distance

OSPF administrative distance.

```
vrouter running config# vrf <vrf> routing ospf distance
```

## all

Default OSPF administrative distance.

```
vrouter running config# vrf <vrf> routing ospf distance  
vrouter running distance# all <uint8>
```

## external

OSPF administrative distance for external routes.

```
vrouter running config# vrf <vrf> routing ospf distance  
vrouter running distance# external <uint8>
```

## inter-area

OSPF administrative distance for inter-area routes.

```
vrouter running config# vrf <vrf> routing ospf distance  
vrouter running distance# inter-area <uint8>
```

## intra-area

OSPF administrative distance for intra-area routes.

```
vrouter running config# vrf <vrf> routing ospf distance  
vrouter running distance# intra-area <uint8>
```

## max-metric

OSPF maximum / infinite-distance metric.

```
vrouter running config# vrf <vrf> routing ospf max-metric
```

## administrative

If true, mark as administratively applied, for an indefinite period.

```
vrouter running config# vrf <vrf> routing ospf max-metric
vrouter running max-metric# administrative true|false
```

## on-shutdown

Advertise stub-router prior to full shutdown of OSPF.

```
vrouter running config# vrf <vrf> routing ospf max-metric
vrouter running max-metric# on-shutdown <uint8>
```

## on-startup

Automatically advertise stub Router-LSA on startup of OSPF.

```
vrouter running config# vrf <vrf> routing ospf max-metric
vrouter running max-metric# on-startup <uint32>
```

## neighbor

Neighbor router.

```
vrouter running config# vrf <vrf> routing ospf
vrouter running ospf# neighbor <neighbor> poll-interval <uint16> priority <uint8>
```

|            |                  |
|------------|------------------|
| <neighbor> | An IPv4 address. |
|------------|------------------|

### poll-interval

Dead neighbor polling interval (in seconds).

```
poll-interval <uint16>
```

### priority

Neighbor priority.

```
priority <uint8>
```

### network

Enable routing on an IP network.

```
vrouter running config# vrf <vrf> routing ospf
vrouter running ospf# network <network> area AREA
```

|           |  |
|-----------|--|
| <network> | An IPv4 prefix: address and CIDR mask. |
|-----------|--|

### area (mandatory)

OSPF area ID.

```
area AREA
```

| AREA values | Description      |
|-------------|------------------|
| <uint32>    | No description.  |
| <A.B.C.D>   | An IPv4 address. |

### passive-interface

Suppress routing updates on an interface.

```
vrouter running config# vrf <vrf> routing ospf
vrouter running ospf# passive-interface <passive-interface> address ADDRESS
```

|                     |                    |
|---------------------|--------------------|
| <passive-interface> | An interface name. |
|---------------------|--------------------|

## address

IPv4 address.

```
address ADDRESS
```

|         |                  |
|---------|------------------|
| ADDRESS | An IPv4 address. |
|---------|------------------|

## timers

Adjust routing timers.

```
vrouter running config# vrf <vrf> routing ospf timers
```

## lsa

Throttling link state advertisement delays.

```
vrouter running config# vrf <vrf> routing ospf timers lsa
```

## min-arrival

Minimum delay in receiving new version of a LSA.

```
vrouter running config# vrf <vrf> routing ospf timers lsa  
vrouter running lsa# min-arrival <uint32>
```

## throttle

Throttling adaptive timer.

```
vrouter running config# vrf <vrf> routing ospf timers throttle
```

## lsa

LSA delay (msec) between transmissions.

```
vrouter running config# vrf <vrf> routing ospf timers throttle  
vrouter running throttle# lsa <uint16>
```

## spf

OSPF SPF timers.

```
vrouter running config# vrf <vrf> routing ospf timers throttle spf
```

## delay (mandatory)

Delay (msec) from first change received till SPF calculation.

```
vrouter running config# vrf <vrf> routing ospf timers throttle spf  
vrouter running spf# delay <uint32>
```

## init-hold-time (mandatory)

Initial hold time (msec) between consecutive SPF calculations.

```
vrouter running config# vrf <vrf> routing ospf timers throttle spf  
vrouter running spf# init-hold-time <uint32>
```

## max-hold-time (mandatory)

Maximum hold time (msec).

```
vrouter running config# vrf <vrf> routing ospf timers throttle spf  
vrouter running spf# max-hold-time <uint32>
```

## distribute-list

Filter networks in routing updates.

```
vrouter running config# vrf <vrf> routing ospf distribute-list out <distribute-list>
```

| <distribute-list> values | Description  |
|--------------------------|--|
| cisco                    | Alternative ABR, cisco implementation.               |
| bgp                      | Border Gateway Protocol (BGP).                       |
| connected                | Connected routes (directly attached subnet or host). |
| kernel                   | Kernel routes (not installed via the zebra RIB).     |
| rip                      | Routing Information Protocol (RIP).                  |
| static                   | Statically configured routes.                        |
| table                    | Non-main Kernel Routing Table.                       |
| nhrp                     | Next Hop Resolution Protocol (NHRP).                 |

## access-list (mandatory)

Access list name.

```
vrouter running config# vrf <vrf> routing ospf distribute-list out <distribute-list>
vrouter running distribute-list out <distribute-list># access-list <string>
```

## redistribute

Redistribute information from another routing protocol.

```
vrouter running config# vrf <vrf> routing ospf
vrouter running ospf# redistribute <redistribute> metric <uint32> metric-type <uint8> \
... route-map <string> id <uint16>
```

| <redistribute> values | Description  |
|-----------------------|--|
| bgp                   | Border Gateway Protocol (BGP).                       |
| connected             | Connected routes (directly attached subnet or host). |
| kernel                | Kernel routes (not installed via the zebra RIB).     |
| ospf                  | Open Shortest Path First.                            |
| rip                   | Routing Information Protocol (RIP).                  |
| static                | Statically configured routes.                        |
| table                 | Non-main Kernel Routing Table.                       |

**metric**

Metric for redistributed routes.

```
metric <uint32>
```

**metric-type**

OSPF exterior metric type for redistributed routes.

```
metric-type <uint8>
```

**route-map**

Route map reference.

```
route-map <string>
```

**id**

Table ID.

```
id <uint16>
```

**rip**

---

**Note:** requires a Turbo Router Network License.

---

RIP router configuration.

```
vrouter running config# vrf <vrf> routing rip
```

## neighbor

Specifies the RIP neighbors. Useful for a non-broadcast multiple access (NBMA) network.

```
vrouter running config# vrf <vrf> routing rip
vrouter running rip# neighbor NEIGHBOR
```

|          |                  |
|----------|------------------|
| NEIGHBOR | An IPv4 address. |
|----------|------------------|

## static-route

RIP static routes.

```
vrouter running config# vrf <vrf> routing rip
vrouter running rip# static-route STATIC-ROUTE
```

|              |  |
|--------------|--|
| STATIC-ROUTE | An IPv4 prefix: address and CIDR mask. |
|--------------|--|

## enabled

Enable or disable router.

```
vrouter running config# vrf <vrf> routing rip
vrouter running rip# enabled true|false
```

### Default value

true

## allow-ecmp

Allow equal-cost multi-path.

```
vrouter running config# vrf <vrf> routing rip
vrouter running rip# allow-ecmp true|false
```

### Default value

false

## default-information-originate

Control distribution of default route.

```
vrouter running config# vrf <vrf> routing rip
vrouter running rip# default-information-originate true|false
```

### Default value

false

## default-metric

Default metric of redistributed routes.

```
vrouter running config# vrf <vrf> routing rip
vrouter running rip# default-metric <uint8>
```

### Default value

1

## network

Enable RIP on the specified IP network.

```
vrouter running config# vrf <vrf> routing rip
vrouter running rip# network NETWORK
```

| NETWORK values | Description                            |
|----------------|--|
| <A.B.C.D/M>    | An IPv4 prefix: address and CIDR mask. |
| <X:X::X:X/M>   | An IPv6 prefix: address and CIDR mask. |

## interface

Enable RIP on the specified interface.

```
vrouter running config# vrf <vrf> routing rip
vrouter running rip# interface INTERFACE
```

|           |                    |
|-----------|--------------------|
| INTERFACE | An interface name. |
|-----------|--------------------|

## passive-interface

A list of interfaces where the sending of RIP packets is disabled.

```
vrouter running config# vrf <vrf> routing rip
vrouter running rip# passive-interface PASSIVE-INTERFACE
```

|                   |                    |
|-------------------|--------------------|
| PASSIVE-INTERFACE | An interface name. |
|-------------------|--------------------|

## administrative-distance

Administrative distance.

```
vrouter running config# vrf <vrf> routing rip administrative-distance
```

## default

Default administrative distance.

```
vrouter running config# vrf <vrf> routing rip administrative-distance
vrouter running administrative-distance# default <uint8>
```

## source

Custom administrative distance per IP prefix.

```
vrouter running config# vrf <vrf> routing rip administrative-distance
vrouter running administrative-distance# source <source> distance <uint8> \
... access-list ACCESS-LIST
```

|          |  |
|----------|--|
| <source> | An IPv4 prefix: address and CIDR mask. |
|----------|--|

## distance (mandatory)

Administrative distance.

```
distance <uint8>
```

## access-list

Access-list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

## redistribute

Redistributes routes learned from other routing protocols.

```
vrouter running config# vrf <vrf> routing rip
vrouter running rip# redistribute <redistribute> metric <uint8> route-map ROUTE-MAP
```

| <redistribute> values | Description  |
|-----------------------|--|
| connected             | Connected routes (directly attached subnet or host). |
| kernel                | Kernel routes (not installed via the zebra RIB).     |
| ospf                  | Open Shortest Path First (OSPFv2).                   |
| bgp                   | Border Gateway Protocol (BGP).                       |
| static                | Statically configured routes.                        |

## metric

Metric used for the redistributed route. If a metric is not specified, the metric configured with the default-metric attribute in RIP router configuration is used. If the default-metric attribute has not been configured, the default metric for redistributed routes is 0.

```
metric <uint8>
```

## route-map

Applies the conditions of the specified route-map to routes that are redistributed into the RIP routing instance.

```
route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

### route-map

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing rip
vrouter running rip# route-map <interface> <route-direction> route-map-name ROUTE-MAP-
↳NAME
```

|             |                    |
|-------------|--------------------|
| <interface> | An interface name. |
|-------------|--------------------|

| <route-direction> values | Description                   |
|--------------------------|-------------------------------|
| in                       | Apply map to incoming routes. |
| out                      | Apply map to outbound routes. |

### route-map-name (mandatory)

Route-map name.

```
route-map-name ROUTE-MAP-NAME
```

|                |                 |
|----------------|-----------------|
| ROUTE-MAP-NAME | Route map name. |
|----------------|-----------------|

### timers

Settings of basic timers.

```
vrouter running config# vrf <vrf> routing rip timers
```

### flush-interval

Interval before a route is flushed from the routing table.

```
vrouter running config# vrf <vrf> routing rip timers
vrouter running timers# flush-interval <uint32>
```

### Default value

120

## holddown-interval

Interval before better routes are released.

```
vrouter running config# vrf <vrf> routing rip timers
vrouter running timers# holddown-interval <uint32>
```

### Default value

180

## update-interval

Interval at which RIP updates are sent.

```
vrouter running config# vrf <vrf> routing rip timers
vrouter running timers# update-interval <uint32>
```

### Default value

30

## version

Set routing protocol version.

```
vrouter running config# vrf <vrf> routing rip version
```

## receive

Advertisement reception - Version control.

```
vrouter running config# vrf <vrf> routing rip version
vrouter running version# receive RECEIVE
```

| RECEIVE values | Description                          |
|----------------|--------------------------------------|
| 1              | Accept RIPv1 updates only.           |
| 2              | Accept RIPv2 updates only.           |
| 1-2            | Accept both RIPv1 and RIPv2 updates. |

### Default value

1-2

## send

Advertisement transmission - Version control.

```
vrouter running config# vrf <vrf> routing rip version
vrouter running version# send SEND
```

| SEND values | Description              |
|-------------|--------------------------|
| 1           | Send RIPv1 updates only. |
| 2           | Send RIPv2 updates only. |

### Default value

2

## distribute-list

Filter networks in routing updates.

```
vrouter running config# vrf <vrf> routing rip
vrouter running rip# distribute-list <interface> <update-direction> access-list ACCESS-
↳LIST \
... prefix-list PREFIX-LIST
```

| <interface> values | Description           |
|--------------------|-----------------------|
| <ifname>           | An interface name.    |
| all                | Match all interfaces. |

| <update-direction> values | Description       |
|---------------------------|-------------------|
| in                        | Incoming updates. |
| out                       | Outgoing updates. |

## access-list

Access-list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

## prefix-list

Prefix-list name.

```
prefix-list PREFIX-LIST
```

|             |                   |
|-------------|-------------------|
| PREFIX-LIST | Prefix list name. |
|-------------|-------------------|

## offset-list

Offset-list to modify route metric.

```
vrouter running config# vrf <vrf> routing rip
vrouter running rip# offset-list <interface> <update-direction> metric <uint8> \
... access-list ACCESS-LIST
```

| <interface> values | Description           |
|--------------------|-----------------------|
| <ifname>           | An interface name.    |
| all                | Match all interfaces. |

| <update-direction> values | Description       |
|---------------------------|-------------------|
| in                        | Incoming updates. |
| out                       | Outgoing updates. |

## metric (mandatory)

Route metric.

```
metric <uint8>
```

## access-list (mandatory)

Access-list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

**state (state only)**

Operational RIP state data.

**route (state only)**

RIP IPv4 route state.

**protocol (state only)**

Route protocol.

```
vrouter> show state vrf <vrf> routing rip state route <route> protocol
```

**route-type (state only)**

Route type.

```
vrouter> show state vrf <vrf> routing rip state route <route> route-type
```

**nexthop (state only)**

Nexthop IPv4 address.

```
vrouter> show state vrf <vrf> routing rip state route <route> nexthop
```

**interface (state only)**

The interface that the route uses.

```
vrouter> show state vrf <vrf> routing rip state route <route> interface
```

### metric (state only)

Route metric.

```
vrouter> show state vrf <vrf> routing rip state route <route> metric
```

### neighbor (state only)

RIP neighbor state.

### last-update (state only)

The time when the most recent RIP update was received from this neighbor.

```
vrouter> show state vrf <vrf> routing rip state neighbor <neighbor> last-update
```

### bad-packets-received (state only)

The number of RIP invalid packets received from this neighbor which were subsequently discarded for any reason (e.g. a version 0 packet, or an unknown command type).

```
vrouter> show state vrf <vrf> routing rip state neighbor <neighbor> bad-packets-  
↪received
```

### bad-routes-received (state only)

The number of routes received from this neighbor, in valid RIP packets, which were ignored for any reason (e.g. unknown address family, or invalid metric).

```
vrouter> show state vrf <vrf> routing rip state neighbor <neighbor> bad-routes-received
```

## ospf6

---

**Note:** requires a Turbo Router Network License.

---

OSPFv3 configuration.

```
vrouter running config# vrf <vrf> routing ospf6
```

**enabled**

Enable or disable OSPFv3.

```
vrouter running config# vrf <vrf> routing ospf6
vrouter running ospf6# enabled true|false
```

**Default value**

true

**router-id**

OSPFv3 router-id in IP address format.

```
vrouter running config# vrf <vrf> routing ospf6
vrouter running ospf6# router-id ROUTER-ID
```

|           |                  |
|-----------|------------------|
| ROUTER-ID | An IPv4 address. |
|-----------|------------------|

**auto-cost**

Calculate OSPF interface cost according to reference bandwidth (Mbits per second).

```
vrouter running config# vrf <vrf> routing ospf6
vrouter running ospf6# auto-cost <uint32>
```

**Default value**

100000

**log-adjacency-changes**

Log changes in adjacency state.

```
vrouter running config# vrf <vrf> routing ospf6
vrouter running ospf6# log-adjacency-changes LOG-ADJACENCY-CHANGES
```

| LOG-ADJACENCY-CHANGES values | Description            |
|------------------------------|------------------------|
| standard                     | Standard logs.         |
| detail                       | Log all state changes. |

## area

OSPFv3 area parameters.

```
vrouter running config# vrf <vrf> routing ospf6 area <area>
```

| <area> values | Description      |
|---------------|------------------|
| <uint32>      | OSPF area ID.    |
| <A.B.C.D>     | An IPv4 address. |

## export-list

Set the filter for networks announced to other areas (access-list name).

```
vrouter running config# vrf <vrf> routing ospf6 area <area>
vrouter running area <area># export-list <string>
```

## import-list

Set the filter for networks from other areas announced to the specified one (access-list name).

```
vrouter running config# vrf <vrf> routing ospf6 area <area>
vrouter running area <area># import-list <string>
```

## stub

Configure area as stub.

```
vrouter running config# vrf <vrf> routing ospf6 area <area> stub
```

## summary

Inject inter-area routes into stub.

```
vrouter running config# vrf <vrf> routing ospf6 area <area> stub
vrouter running stub# summary true|false
```

## Default value

true

## filter-list

Filter networks between areas.

```
vrouter running config# vrf <vrf> routing ospf6 area <area> filter-list
```

## input

Filter networks sent to this area (prefix-list name).

```
vrouter running config# vrf <vrf> routing ospf6 area <area> filter-list
vrouter running filter-list# input <string>
```

## output

Filter networks sent from this area (prefix-list name).

```
vrouter running config# vrf <vrf> routing ospf6 area <area> filter-list
vrouter running filter-list# output <string>
```

## range

Summarize routes matching address/mask (border routers only).

```
vrouter running config# vrf <vrf> routing ospf6 area <area>
vrouter running area <area># range <range> advertise true|false cost <uint32>
```

|         |  |
|---------|--|
| <range> | An IPv6 prefix: address and CIDR mask. |
|---------|--|

## advertise

Advertise this range.

```
advertise true|false
```

## Default value

true

### cost

User specified metric for this range.

```
cost <uint32>
```

### distance

OSPF administrative distance.

```
vrouter running config# vrf <vrf> routing ospf6 distance
```

### all

Default OSPF administrative distance.

```
vrouter running config# vrf <vrf> routing ospf6 distance  
vrouter running distance# all <uint8>
```

### external

OSPF administrative distance for external routes.

```
vrouter running config# vrf <vrf> routing ospf6 distance  
vrouter running distance# external <uint8>
```

### inter-area

OSPF administrative distance for inter-area routes.

```
vrouter running config# vrf <vrf> routing ospf6 distance  
vrouter running distance# inter-area <uint8>
```

## intra-area

OSPF administrative distance for intra-area routes.

```
vrouter running config# vrf <vrf> routing ospf6 distance
vrouter running distance# intra-area <uint8>
```

## interface

Enable routing on an IPv6 interface.

```
vrouter running config# vrf <vrf> routing ospf6
vrouter running ospf6# interface <interface> area AREA
```

|             |                    |
|-------------|--------------------|
| <interface> | An interface name. |
|-------------|--------------------|

## area (mandatory)

OSPF6 area ID.

```
area AREA
```

|      |                  |
|------|------------------|
| AREA | An IPv4 address. |
|------|------------------|

## redistribute

Redistribute information from another routing protocol.

```
vrouter running config# vrf <vrf> routing ospf6
vrouter running ospf6# redistribute <redistribute> route-map <string>
```

| <redistribute> values | Description  |
|-----------------------|--|
| babel                 | Babel routing protocol (Babel).                              |
| bgp                   | Border Gateway Protocol (BGP).                               |
| connected             | Connected routes (directly attached subnet or host).         |
| kernel                | Kernel routes (not installed via the zebra RIB).             |
| ripng                 | Routing Information Protocol next-generation (IPv6) (RIPng). |
| static                | Statically configured routes.                                |
| table                 | Non-main Kernel Routing Table.                               |

## route-map

Route map reference.

```
route-map <string>
```

## timers

Adjust routing timers.

```
vrouter running config# vrf <vrf> routing ospf6 timers
```

## lsa

Throttling link state advertisement delays.

```
vrouter running config# vrf <vrf> routing ospf6 timers lsa
```

## min-arrival

Minimum delay in receiving new version of a LSA.

```
vrouter running config# vrf <vrf> routing ospf6 timers lsa  
vrouter running lsa# min-arrival <uint32>
```

## throttle

Throttling adaptive timer.

```
vrouter running config# vrf <vrf> routing ospf6 timers throttle
```

## lsa

LSA delay (msec) between transmissions.

```
vrouter running config# vrf <vrf> routing ospf6 timers throttle  
vrouter running throttle# lsa <uint16>
```

## spf

OSPF SPF timers.

```
vrouter running config# vrf <vrf> routing ospf6 timers throttle spf
```

### delay (mandatory)

Delay (msec) from first change received till SPF calculation.

```
vrouter running config# vrf <vrf> routing ospf6 timers throttle spf
vrouter running spf# delay <uint32>
```

### init-hold-time (mandatory)

Initial hold time (msec) between consecutive SPF calculations.

```
vrouter running config# vrf <vrf> routing ospf6 timers throttle spf
vrouter running spf# init-hold-time <uint32>
```

### max-hold-time (mandatory)

Maximum hold time (msec).

```
vrouter running config# vrf <vrf> routing ospf6 timers throttle spf
vrouter running spf# max-hold-time <uint32>
```

## ripng

---

**Note:** requires a Turbo Router Network License.

---

RIPng router configuration.

```
vrouter running config# vrf <vrf> routing ripng
```

## aggregate

Set aggregate RIPng route announcement.

```
vrouter running config# vrf <vrf> routing ripng
vrouter running ripng# aggregate AGGREGATE
```

|           |  |
|-----------|--|
| AGGREGATE | An IPv6 prefix: address and CIDR mask. |
|-----------|--|

## static-route

RIPng static routes.

```
vrouter running config# vrf <vrf> routing ripng
vrouter running ripng# static-route STATIC-ROUTE
```

|              |  |
|--------------|--|
| STATIC-ROUTE | An IPv6 prefix: address and CIDR mask. |
|--------------|--|

## enabled

Enable or disable router.

```
vrouter running config# vrf <vrf> routing ripng
vrouter running ripng# enabled true|false
```

### Default value

true

## allow-ecmp

Allow equal-cost multi-path.

```
vrouter running config# vrf <vrf> routing ripng
vrouter running ripng# allow-ecmp true|false
```

### Default value

false

## default-information-originate

Control distribution of default route.

```
vrouter running config# vrf <vrf> routing ripng
vrouter running ripng# default-information-originate true|false
```

### Default value

false

## default-metric

Default metric of redistributed routes.

```
vrouter running config# vrf <vrf> routing ripng
vrouter running ripng# default-metric <uint8>
```

### Default value

1

## network

Enable RIP on the specified IP network.

```
vrouter running config# vrf <vrf> routing ripng
vrouter running ripng# network NETWORK
```

| NETWORK values | Description                            |
|----------------|--|
| <A.B.C.D/M>    | An IPv4 prefix: address and CIDR mask. |
| <X:X::X:X/M>   | An IPv6 prefix: address and CIDR mask. |

## interface

Enable RIP on the specified interface.

```
vrouter running config# vrf <vrf> routing ripng
vrouter running ripng# interface INTERFACE
```

|           |                    |
|-----------|--------------------|
| INTERFACE | An interface name. |
|-----------|--------------------|

## passive-interface

A list of interfaces where the sending of RIP packets is disabled.

```
vrouter running config# vrf <vrf> routing ripng
vrouter running ripng# passive-interface PASSIVE-INTERFACE
```

|                   |                    |
|-------------------|--------------------|
| PASSIVE-INTERFACE | An interface name. |
|-------------------|--------------------|

## redistribute

Redistributes routes learned from other routing protocols.

```
vrouter running config# vrf <vrf> routing ripng
vrouter running ripng# redistribute <redistribute> metric <uint8> route-map ROUTE-MAP
```

| <redistribute> values | Description  |
|-----------------------|--|
| connected             | Connected routes (directly attached subnet or host). |
| kernel                | Kernel routes (not installed via the zebra RIB).     |
| ospf6                 | Open Shortest Path First (OSPFv3).                   |
| bgp                   | Border Gateway Protocol (BGP).                       |
| static                | Statically configured routes.                        |

## metric

Metric used for the redistributed route. If a metric is not specified, the metric configured with the default-metric attribute in RIPng router configuration is used. If the default-metric attribute has not been configured, the default metric for redistributed routes is 0.

```
metric <uint8>
```

## route-map

Applies the conditions of the specified route-map to routes that are redistributed into the RIPng routing instance.

```
route-map ROUTE-MAP
```

|           |                 |
|-----------|-----------------|
| ROUTE-MAP | Route map name. |
|-----------|-----------------|

## timers

Settings of basic timers.

```
vrouter running config# vrf <vrf> routing ripng timers
```

## flush-interval

Interval before a route is flushed from the routing table.

```
vrouter running config# vrf <vrf> routing ripng timers  
vrouter running timers# flush-interval <uint16>
```

## Default value

120

## holddown-interval

Interval before better routes are released.

```
vrouter running config# vrf <vrf> routing ripng timers  
vrouter running timers# holddown-interval <uint16>
```

## Default value

180

## update-interval

Interval at which RIP updates are sent.

```
vrouter running config# vrf <vrf> routing ripng timers  
vrouter running timers# update-interval <uint16>
```

## Default value

30

### distributed-list

Filter networks in routing updates.

```

vrouter running config# vrf <vrf> routing ripng
vrouter running ripng# distributed-list <interface> <update-direction> access-list_
↳ACCESS-LIST \
... prefix-list PREFIX-LIST

```

| <interface> values | Description           |
|--------------------|-----------------------|
| <ifname>           | An interface name.    |
| all                | Match all interfaces. |

| <update-direction> values | Description       |
|---------------------------|-------------------|
| in                        | Incoming updates. |
| out                       | Outgoing updates. |

### access-list

Access-list name.

```

access-list ACCESS-LIST

```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

### prefix-list

Prefix-list name.

```

prefix-list PREFIX-LIST

```

|             |                   |
|-------------|-------------------|
| PREFIX-LIST | Prefix list name. |
|-------------|-------------------|

### offset-list

Offset-list to modify route metric.

```
vrouter running config# vrf <vrf> routing ripng
vrouter running ripng# offset-list <interface> <update-direction> metric <uint8> \
... access-list ACCESS-LIST
```

| <interface> values | Description           |
|--------------------|-----------------------|
| <ifname>           | An interface name.    |
| all                | Match all interfaces. |

| <update-direction> values | Description       |
|---------------------------|-------------------|
| in                        | Incoming updates. |
| out                       | Outgoing updates. |

### metric (mandatory)

Route metric.

```
metric <uint8>
```

### access-list (mandatory)

Access-list name.

```
access-list ACCESS-LIST
```

|             |                   |
|-------------|-------------------|
| ACCESS-LIST | Access list name. |
|-------------|-------------------|

### state (state only)

Operational RIPng state data.

**route (state only)**

RIPng IPv6 route state.

**protocol (state only)**

Route protocol.

```
vrouter> show state vrf <vrf> routing ripng state route <route> protocol
```

**route-type (state only)**

Route type.

```
vrouter> show state vrf <vrf> routing ripng state route <route> route-type
```

**nexthop (state only)**

Nexthop IPv6 address.

```
vrouter> show state vrf <vrf> routing ripng state route <route> nexthop
```

**metric (state only)**

Route metric.

```
vrouter> show state vrf <vrf> routing ripng state route <route> metric
```

**neighbor (state only)**

RIP neighbor state.

### last-update (state only)

The time when the most recent RIP update was received from this neighbor.

```
vrouter> show state vrf <vrf> routing ripng state neighbor <neighbor> last-update
```

### bad-packets-received (state only)

The number of RIP invalid packets received from this neighbor which were subsequently discarded for any reason (e.g. a version 0 packet, or an unknown command type).

```
vrouter> show state vrf <vrf> routing ripng state neighbor <neighbor> bad-packets-  
↪received
```

### bad-routes-received (state only)

The number of routes received from this neighbor, in valid RIP packets, which were ignored for any reason (e.g. unknown address family, or invalid metric).

```
vrouter> show state vrf <vrf> routing ripng state neighbor <neighbor> bad-routes-  
↪received
```

## policy-based-routing

---

**Note:** requires a Turbo Router Network License.

---

Configure the policy-based routing.

```
vrouter running config# vrf <vrf> routing policy-based-routing
```

### ipv4-rule

Configure an IPv4 rule.

```
vrouter running config# vrf <vrf> routing policy-based-routing  
vrouter running policy-based-routing# ipv4-rule <0-99999> [not] \  
... match inbound-interface INBOUND-INTERFACE mark MARK source SOURCE destination_  
↪DESTINATION \  
... action lookup LOOKUP
```

<0-99999> Priority of the rule. High number means lower priority.

**not**

Invert the match.

not

**match**

Configure the packet selector.

match inbound-interface INBOUND-INTERFACE mark MARK source SOURCE destination DESTINATION

**inbound-interface**

Match this incoming interface.

inbound-interface INBOUND-INTERFACE

INBOUND-INTERFACE An interface name.

**mark**

Match this mark filter.

mark MARK

| MARK values                     | Description           |
|---------------------------------|-----------------------|
| <0x0-0xffffffff>                | Firewall mark.        |
| <0x0-0xffffffff/0x0-0xffffffff> | Firewall mark filter. |

**source**

Match this source address or prefix.

```
source SOURCE
```

| SOURCE values | Description                            |
|---------------|--|
| <A.B.C.D>     | An IPv4 address.                       |
| <A.B.C.D/M>   | An IPv4 prefix: address and CIDR mask. |

**destination**

Match this destination address or prefix.

```
destination DESTINATION
```

| DESTINATION values | Description                            |
|--------------------|--|
| <A.B.C.D>          | An IPv4 address.                       |
| <A.B.C.D/M>        | An IPv4 prefix: address and CIDR mask. |

**outbound-interface (state only)**

Match this outgoing interface.

```
vrouter> show state vrf <vrf> routing policy-based-routing ipv4-rule <0-99999> match_
↳outbound-interface
```

**tos (state only)**

Match this tos.

```
vrouter> show state vrf <vrf> routing policy-based-routing ipv4-rule <0-99999> match_
↳tos
```

**other (state only)**

Match a specific attribute.

**value (state only)**

The value to match.

```
vrouter> show state vrf <vrf> routing policy-based-routing ipv4-rule <0-99999> match.
↳other <string> value
```

**action**

Configure the action for packets matching the selector.

```
action lookup LOOKUP
```

**lookup (mandatory)**

Lookup in this table.

```
lookup LOOKUP
```

| LOOKUP values | Description   |
|---------------|---|
| <uint32>      | Table type.   |
| local         | High priority control routes for local and broadcast addresses (table 255).                     |
| main          | Normal routing table, containing all non-policy routes (table 254).                             |
| default       | Reserved for some post-processing if no previous default rules selected the packet (table 253). |

**goto (state only)**

Jump to the specified priority rule.

```
vrouter> show state vrf <vrf> routing policy-based-routing ipv4-rule <0-99999> action.
↳goto
```

## other (state only)

Other actions.

```
vrouter> show state vrf <vrf> routing policy-based-routing ipv4-rule <0-99999> action.
↳other
```

## ipv6-rule

Configure an IPv6 rule.

```
vrouter running config# vrf <vrf> routing policy-based-routing
vrouter running policy-based-routing# ipv6-rule <0-99999> [not] \
... match inbound-interface INBOUND-INTERFACE mark MARK source SOURCE destination.
↳DESTINATION \
... action lookup LOOKUP
```

|           |   |
|-----------|---|
| <0-99999> | Priority of the rule. High number means lower priority. |
|-----------|---|

## not

Invert the match.

```
not
```

## match

Configure the packet selector.

```
match inbound-interface INBOUND-INTERFACE mark MARK source SOURCE destination.
↳DESTINATION
```

## inbound-interface

Match this incoming interface.

```
inbound-interface INBOUND-INTERFACE
```

|                   |                    |
|-------------------|--------------------|
| INBOUND-INTERFACE | An interface name. |
|-------------------|--------------------|

**mark**

Match this mark filter.

```
mark MARK
```

| MARK values                     | Description           |
|---------------------------------|-----------------------|
| <0x0-0xffffffff>                | Firewall mark.        |
| <0x0-0xffffffff/0x0-0xffffffff> | Firewall mark filter. |

**source**

Match this source address or prefix.

```
source SOURCE
```

| SOURCE values | Description                            |
|---------------|--|
| <X:X::X:X>    | An IPv6 address.                       |
| <X:X::X:X/M>  | An IPv6 prefix: address and CIDR mask. |

**destination**

Match this destination address or prefix.

```
destination DESTINATION
```

| DESTINATION values | Description                            |
|--------------------|--|
| <X:X::X:X>         | An IPv6 address.                       |
| <X:X::X:X/M>       | An IPv6 prefix: address and CIDR mask. |

**outbound-interface (state only)**

Match this outgoing interface.

```
vrouter> show state vrf <vrf> routing policy-based-routing ipv6-rule <0-99999> match_
↳outbound-interface
```

**tos (state only)**

Match this tos.

```
vrouter> show state vrf <vrf> routing policy-based-routing ipv6-rule <0-99999> match_
↳tos
```

**other (state only)**

Match a specific attribute.

**value (state only)**

The value to match.

```
vrouter> show state vrf <vrf> routing policy-based-routing ipv6-rule <0-99999> match_
↳other <string> value
```

**action**

Configure the action for packets matching the selector.

```
action lookup LOOKUP
```

**lookup (mandatory)**

Lookup in this table.

```
lookup LOOKUP
```

| LOOKUP values | Description   |
|---------------|---|
| <uint32>      | Table type.   |
| local         | High priority control routes for local and broadcast addresses (table 255).                     |
| main          | Normal routing table, containing all non-policy routes (table 254).                             |
| default       | Reserved for some post-processing if no previous default rules selected the packet (table 253). |

**goto (state only)**

Goto to the specified priority rule.

```
vrouter> show state vrf <vrf> routing policy-based-routing ipv6-rule <0-99999> action.  
↳goto
```

**other (state only)**

Other actions.

```
vrouter> show state vrf <vrf> routing policy-based-routing ipv6-rule <0-99999> action.  
↳other
```

**rib (state only)**

Routing information base.

**ipv4-count (state only)**

IPv4 routes statistics.

**kernel (state only)**

Kernel routes count.

**routes (state only)**

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count kernel routes
```

**installed-routes (state only)**

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count kernel installed-routes
```

**connected (state only)**

Connected routes count.

**routes (state only)**

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count connected routes
```

**installed-routes (state only)**

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count connected installed-routes
```

**static (state only)**

Static routes count.

**routes (state only)**

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count static routes
```

**installed-routes (state only)**

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count static installed-routes
```

**ospf (state only)**

OSPF routes count.

**routes (state only)**

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count ospf routes
```

**installed-routes (state only)**

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count ospf installed-routes
```

**ebgp (state only)**

EBGP routes count.

**routes (state only)**

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count ebgp routes
```

### installed-routes (state only)

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count ebgp installed-routes
```

### ibgp (state only)

IBGP routes count.

### routes (state only)

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count ibgp routes
```

### installed-routes (state only)

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count ibgp installed-routes
```

### nhrp (state only)

NHRP routes count.

### routes (state only)

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count nhrp routes
```

### installed-routes (state only)

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count nhrp installed-routes
```

### total (state only)

Total routes count.

### routes (state only)

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count total routes
```

### installed-routes (state only)

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv4-count total installed-routes
```

### ipv6-count (state only)

IPv6 routes statistics.

### kernel (state only)

Kernel routes count.

### routes (state only)

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count kernel routes
```

### installed-routes (state only)

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count kernel installed-routes
```

### connected (state only)

Connected routes count.

### routes (state only)

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count connected routes
```

### installed-routes (state only)

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count connected installed-routes
```

### static (state only)

Static routes count.

### routes (state only)

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count static routes
```

**installed-routes (state only)**

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count static installed-routes
```

**ospf (state only)**

OSPF routes count.

**routes (state only)**

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count ospf routes
```

**installed-routes (state only)**

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count ospf installed-routes
```

**ebgp (state only)**

EBGP routes count.

**routes (state only)**

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count ebgp routes
```

**installed-routes (state only)**

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count ebgp installed-routes
```

**ibgp (state only)**

IBGP routes count.

**routes (state only)**

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count ibgp routes
```

**installed-routes (state only)**

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count ibgp installed-routes
```

**nhrp (state only)**

NHRP routes count.

**routes (state only)**

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count nhrp routes
```

**installed-routes (state only)**

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count nhrp installed-routes
```

**total (state only)**

Total routes count.

**routes (state only)**

Number of routes in RIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count total routes
```

**installed-routes (state only)**

Number of routes in FIB.

```
vrouter> show state vrf <vrf> routing rib ipv6-count total installed-routes
```

**ipv4-route (state only)**

IPv4 routes in RIB.

**next-hop (state only)**

Route next-hops.

**protocol (state only)**

Route protocol.

```
vrouter> show state vrf <vrf> routing rib ipv4-route <ipv4-route> next-hop <next-hop> ↵  
↵-protocol
```

### distance (state only)

Distance value for this route.

```
vrouter> show state vrf <vrf> routing rib ipv4-route <ipv4-route> next-hop <next-hop>_  
↳distance
```

### metric (state only)

Route metric.

```
vrouter> show state vrf <vrf> routing rib ipv4-route <ipv4-route> next-hop <next-hop>_  
↳metric
```

### interface (state only)

Output interface.

```
vrouter> show state vrf <vrf> routing rib ipv4-route <ipv4-route> next-hop <next-hop>_  
↳interface
```

### selected (state only)

If true, route is selected.

```
vrouter> show state vrf <vrf> routing rib ipv4-route <ipv4-route> next-hop <next-hop>_  
↳selected
```

### fib (state only)

If true, route is in Forwarding Information Base.

```
vrouter> show state vrf <vrf> routing rib ipv4-route <ipv4-route> next-hop <next-hop>_  
↳fib
```

### directly-connected (state only)

If true, route is directly connected.

```
vrouter> show state vrf <vrf> routing rib ipv4-route <ipv4-route> next-hop <next-hop>_  
↳directly-connected
```

### duplicate (state only)

If true, route is duplicate.

```
vrouter> show state vrf <vrf> routing rib ipv4-route <ipv4-route> next-hop <next-hop>_  
↳duplicate
```

### active (state only)

If true, route is active.

```
vrouter> show state vrf <vrf> routing rib ipv4-route <ipv4-route> next-hop <next-hop>_  
↳active
```

### on-link (state only)

If true, on link is set.

```
vrouter> show state vrf <vrf> routing rib ipv4-route <ipv4-route> next-hop <next-hop>_  
↳on-link
```

### recursive (state only)

If true, recursive is set.

```
vrouter> show state vrf <vrf> routing rib ipv4-route <ipv4-route> next-hop <next-hop>_  
↳recursive
```

### uptime (state only)

Route uptime.

```
vrouter> show state vrf <vrf> routing rib ipv4-route <ipv4-route> next-hop <next-hop>_
↳uptime
```

### ipv6-route (state only)

IPv6 routes in RIB.

### next-hop (state only)

Route next-hops.

### protocol (state only)

Route protocol.

```
vrouter> show state vrf <vrf> routing rib ipv6-route <ipv6-route> next-hop <next-hop>_
↳protocol
```

### distance (state only)

Distance value for this route.

```
vrouter> show state vrf <vrf> routing rib ipv6-route <ipv6-route> next-hop <next-hop>_
↳distance
```

### metric (state only)

Route metric.

```
vrouter> show state vrf <vrf> routing rib ipv6-route <ipv6-route> next-hop <next-hop>_
↳metric
```

### interface (state only)

Output interface.

```
vrouter> show state vrf <vrf> routing rib ipv6-route <ipv6-route> next-hop <next-hop>_  
↳interface
```

### selected (state only)

If true, route is selected.

```
vrouter> show state vrf <vrf> routing rib ipv6-route <ipv6-route> next-hop <next-hop>_  
↳selected
```

### fib (state only)

If true, route is in Forwarding Information Base.

```
vrouter> show state vrf <vrf> routing rib ipv6-route <ipv6-route> next-hop <next-hop>_  
↳fib
```

### directly-connected (state only)

If true, route is directly connected.

```
vrouter> show state vrf <vrf> routing rib ipv6-route <ipv6-route> next-hop <next-hop>_  
↳directly-connected
```

### duplicate (state only)

If true, route is duplicate.

```
vrouter> show state vrf <vrf> routing rib ipv6-route <ipv6-route> next-hop <next-hop>_  
↳duplicate
```

### active (state only)

If true, route is active.

```
vrouter> show state vrf <vrf> routing rib ipv6-route <ipv6-route> next-hop <next-hop>_  
↳active
```

### on-link (state only)

If true, on link is set.

```
vrouter> show state vrf <vrf> routing rib ipv6-route <ipv6-route> next-hop <next-hop>_  
↳on-link
```

### recursive (state only)

If true, recursive is set.

```
vrouter> show state vrf <vrf> routing rib ipv6-route <ipv6-route> next-hop <next-hop>_  
↳recursive
```

### uptime (state only)

Route uptime.

```
vrouter> show state vrf <vrf> routing rib ipv6-route <ipv6-route> next-hop <next-hop>_  
↳uptime
```

## 3.2.29 DHCP

### server

DHCP server configuration.

```
vrouter running config# vrf <vrf> dhcp server
```

## enabled

Enable/Disable DHCP server on this VRF.

```
vrouter running config# vrf <vrf> dhcp server
vrouter running server# enabled true|false
```

### Default value

true

## default-lease-time

Default network address lease time assigned to DHCP clients (in seconds, at least 180s).

```
vrouter running config# vrf <vrf> dhcp server
vrouter running server# default-lease-time <uint32>
```

### Default value

43200

## max-lease-time

Maximum network address lease time assigned to DHCP clients (in seconds, at least 180s or the default-lease value).

```
vrouter running config# vrf <vrf> dhcp server
vrouter running server# max-lease-time <uint32>
```

### Default value

86400

## dhcp-options

Default DHCP options configuration.

```
vrouter running config# vrf <vrf> dhcp server dhcp-options
```

### dhcp-server-identifier

DHCP server identifier (IPv4 address) used in DHCP messages to allow the client to distinguish between lease offers.

```
vrouter running config# vrf <vrf> dhcp server dhcp-options
vrouter running dhcp-options# dhcp-server-identifier DHCP-SERVER-IDENTIFIER
```

|                        |                  |
|------------------------|------------------|
| DHCP-SERVER-IDENTIFIER | An IPv4 address. |
|------------------------|------------------|

### domain-name

Name of the domain.

```
vrouter running config# vrf <vrf> dhcp server dhcp-options
vrouter running dhcp-options# domain-name <string>
```

### domain-name-server

Domain name server (IPv4 address) listed in order of preference.

```
vrouter running config# vrf <vrf> dhcp server dhcp-options
vrouter running dhcp-options# domain-name-server DOMAIN-NAME-SERVER
```

|                    |                  |
|--------------------|------------------|
| DOMAIN-NAME-SERVER | An IPv4 address. |
|--------------------|------------------|

### ntp-server

NTP server (IPv4 address) listed in order of preference.

```
vrouter running config# vrf <vrf> dhcp server dhcp-options
vrouter running dhcp-options# ntp-server NTP-SERVER
```

|            |                  |
|------------|------------------|
| NTP-SERVER | An IPv4 address. |
|------------|------------------|

## interface-mtu

Minimum Transmission Unit (MTU) of the interface.

```
vrouter running config# vrf <vrf> dhcp server dhcp-options
vrouter running dhcp-options# interface-mtu <uint16>
```

## netbios-name-server

NETBIOS name server listed in order of preference.

```
vrouter running config# vrf <vrf> dhcp server dhcp-options
vrouter running dhcp-options# netbios-name-server NETBIOS-NAME-SERVER
```

| NETBIOS-NAME-SERVER values | Description      |
|----------------------------|------------------|
| <A.B.C.D>                  | An IPv4 address. |
| <X:X::X:X>                 | An IPv6 address. |

## netbios-node-type

NETBIOS node type.

```
vrouter running config# vrf <vrf> dhcp server dhcp-options
vrouter running dhcp-options# netbios-node-type NETBIOS-NODE-TYPE
```

| NETBIOS-NODE-TYPE values | Description                    |
|--------------------------|--------------------------------|
| B-node                   | Broadcast - no WINS.           |
| P-node                   | Peer - WINS only.              |
| M-node                   | Mixed - broadcast, then WINS.  |
| H-node                   | Hybrid - WINS, then broadcast. |

## netbios-scope

NETBIOS over TCP/IP scope parameter for the client as specified in RFC 1001/1002.

```
vrouter running config# vrf <vrf> dhcp server dhcp-options
vrouter running dhcp-options# netbios-scope <string>
```

## time-offset

Time offset in seconds from UTC.

```
vrouter running config# vrf <vrf> dhcp server dhcp-options  
vrouter running dhcp-options# time-offset <int32>
```

## subnet

Subnet configuration.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet>
```

|          |  |
|----------|--|
| <subnet> | An IPv4 prefix: address and CIDR mask. |
|----------|--|

## interface

Interface on which the DHCP server should listen.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet>  
vrouter running subnet <subnet># interface INTERFACE
```

|           |                    |
|-----------|--------------------|
| INTERFACE | An interface name. |
|-----------|--------------------|

## default-gateway

IPv4 address of the gateway listed in order of preference.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet>  
vrouter running subnet <subnet># default-gateway DEFAULT-GATEWAY
```

|                 |                  |
|-----------------|------------------|
| DEFAULT-GATEWAY | An IPv4 address. |
|-----------------|------------------|

### default-lease-time

Default network address lease time assigned to DHCP clients for this subnet (in seconds, at least 180s).

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet>
vrouter running subnet <subnet># default-lease-time <uint32>
```

### max-lease-time

Maximum network address lease time assigned to DHCP clients for this subnet (in seconds, at least 180s or the default-lease value).

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet>
vrouter running subnet <subnet># max-lease-time <uint32>
```

### state (state only)

Subnet state.

```
vrouter> show state vrf <vrf> dhcp server subnet <subnet> state
```

### range

IPv4 range.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet>
vrouter running subnet <subnet># range <start-ip> <end-ip>
```

|            |                  |
|------------|------------------|
| <start-ip> | An IPv4 address. |
|------------|------------------|

|          |                  |
|----------|------------------|
| <end-ip> | An IPv4 address. |
|----------|------------------|

## host

Mapping from MAC address to IP address.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet>
vrouter running subnet <subnet># host <string> MAC-ADDRESS IP-ADDRESS
```

|          |   |
|----------|---|
| <string> | Host name for static MAC to IP address mapping. |
|----------|---|

### MAC-ADDRESS (mandatory)

MAC address of the host.

|             |
|-------------|
| MAC-ADDRESS |
|-------------|

|             |                          |
|-------------|--------------------------|
| MAC-ADDRESS | An IEEE 802 MAC address. |
|-------------|--------------------------|

### IP-ADDRESS (mandatory)

IPv4 address of the host.

|            |
|------------|
| IP-ADDRESS |
|------------|

|            |                  |
|------------|------------------|
| IP-ADDRESS | An IPv4 address. |
|------------|------------------|

## dhcp-options

DHCP options specific to this subnet.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet> dhcp-options
```

### dhcp-server-identifier

DHCP server identifier (IPv4 address) used in DHCP messages to allow the client to distinguish between lease offers.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet> dhcp-options
vrouter running dhcp-options# dhcp-server-identifier DHCP-SERVER-IDENTIFIER
```

|                        |                  |
|------------------------|------------------|
| DHCP-SERVER-IDENTIFIER | An IPv4 address. |
|------------------------|------------------|

### domain-name

Name of the domain.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet> dhcp-options
vrouter running dhcp-options# domain-name <string>
```

### domain-name-server

Domain name server (IPv4 address) listed in order of preference.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet> dhcp-options
vrouter running dhcp-options# domain-name-server DOMAIN-NAME-SERVER
```

|                    |                  |
|--------------------|------------------|
| DOMAIN-NAME-SERVER | An IPv4 address. |
|--------------------|------------------|

### ntp-server

NTP server (IPv4 address) listed in order of preference.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet> dhcp-options
vrouter running dhcp-options# ntp-server NTP-SERVER
```

|            |                  |
|------------|------------------|
| NTP-SERVER | An IPv4 address. |
|------------|------------------|

### interface-mtu

Minimum Transmission Unit (MTU) of the interface.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet> dhcp-options
vrouter running dhcp-options# interface-mtu <uint16>
```

## netbios-name-server

NETBIOS name server listed in order of preference.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet> dhcp-options
vrouter running dhcp-options# netbios-name-server NETBIOS-NAME-SERVER
```

| NETBIOS-NAME-SERVER values | Description      |
|----------------------------|------------------|
| <A.B.C.D>                  | An IPv4 address. |
| <X:X::X:X>                 | An IPv6 address. |

## netbios-node-type

NETBIOS node type.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet> dhcp-options
vrouter running dhcp-options# netbios-node-type NETBIOS-NODE-TYPE
```

| NETBIOS-NODE-TYPE values | Description                    |
|--------------------------|--------------------------------|
| B-node                   | Broadcast - no WINS.           |
| P-node                   | Peer - WINS only.              |
| M-node                   | Mixed - broadcast, then WINS.  |
| H-node                   | Hybrid - WINS, then broadcast. |

## netbios-scope

NETBIOS over TCP/IP scope parameter for the client as specified in RFC 1001/1002.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet> dhcp-options
vrouter running dhcp-options# netbios-scope <string>
```

## time-offset

Time offset in seconds from UTC.

```
vrouter running config# vrf <vrf> dhcp server subnet <subnet> dhcp-options
vrouter running dhcp-options# time-offset <int32>
```

### dhcp-server-leases (state only)

State of leases for DHCP server.

#### starts (state only)

Lease start time.

```
vrouter> show state vrf <vrf> dhcp server dhcp-server-leases <dhcp-server-leases> ↵  
↪starts
```

#### ends (state only)

Lease end time.

```
vrouter> show state vrf <vrf> dhcp server dhcp-server-leases <dhcp-server-leases> ends
```

#### hw-mac-address (state only)

MAC address of the network interface on which the lease will be used.

```
vrouter> show state vrf <vrf> dhcp server dhcp-server-leases <dhcp-server-leases> hw-  
↪mac-address
```

#### uid (state only)

Client identifier used by the client to acquire the lease.

```
vrouter> show state vrf <vrf> dhcp server dhcp-server-leases <dhcp-server-leases> uid
```

#### client-hostname (state only)

Client host name sent using client-hostname statement.

```
vrouter> show state vrf <vrf> dhcp server dhcp-server-leases <dhcp-server-leases> ↵  
↪client-hostname
```

### binding-state (state only)

Lease's binding state.

```
vrouter> show state vrf <vrf> dhcp server dhcp-server-leases <dhcp-server-leases> ↵  
↪binding-state
```

### next-binding-state (state only)

State the lease will move to when the current state expires.

```
vrouter> show state vrf <vrf> dhcp server dhcp-server-leases <dhcp-server-leases> next-  
↪binding-state
```

### option-agent-circuit-id (state only)

Circuit ID option sent by the relay agent.

```
vrouter> show state vrf <vrf> dhcp server dhcp-server-leases <dhcp-server-leases> ↵  
↪option-agent-circuit-id
```

### option-agent-remote-id (state only)

Remote ID option sent by the relay agent.

```
vrouter> show state vrf <vrf> dhcp server dhcp-server-leases <dhcp-server-leases> ↵  
↪option-agent-remote-id
```

### vendor-class-identifier (state only)

Client-supplied Vendor Class Identifier option.

```
vrouter> show state vrf <vrf> dhcp server dhcp-server-leases <dhcp-server-leases> ↵  
↪vendor-class-identifier
```

## relay

DHCP relay configuration.

```
vrouter running config# vrf <vrf> dhcp relay
```

## enabled

Enable/Disable DHCP relay on this VRF.

```
vrouter running config# vrf <vrf> dhcp relay
vrouter running relay# enabled true|false
```

## Default value

true

## handle-option

Handling of DHCPv4 packets that already contain relay agent options.

```
vrouter running config# vrf <vrf> dhcp relay
vrouter running relay# handle-option HANDLE-OPTION
```

| HANDLE-OPTION values | Description  |
|----------------------|--|
| append               | Append our own set of relay options to the packet, leaving the supplied option field intact. |
| replace              | Replace the existing agent option field.   |
| forward              | Forward the packet unchanged.  |
| discard              | Discard the packet.  |

## Default value

append

## drop-unmatched

If true, drop packets from upstream servers if they were generated in response to a different relay agent.

```
vrouter running config# vrf <vrf> dhcp relay
vrouter running relay# drop-unmatched true|false
```

### Default value

false

## hop-count

Maximum hop count before packets are discarded.

```
vrouter running config# vrf <vrf> dhcp relay
vrouter running relay# hop-count <0-255>
```

### Default value

10

## max-size

Maximum packet size to send to a DHCPv4 server. If a DHCP packet size surpasses this value it will be forwarded without appending relay agent information.

```
vrouter running config# vrf <vrf> dhcp relay
vrouter running relay# max-size <64-1400>
```

### Default value

576

## dhcp-server

Configuration of DHCP server to which DHCP queries should be relayed.

```
vrouter running config# vrf <vrf> dhcp relay dhcp-server <dhcp-server>
```

|               |                  |
|---------------|------------------|
| <dhcp-server> | An IPv4 address. |
|---------------|------------------|

## enabled

Enable/Disable DHCP relay for this server.

```
vrouter running config# vrf <vrf> dhcp relay dhcp-server <dhcp-server>
vrouter running dhcp-server <dhcp-server># enabled true|false
```

### Default value

true

## interface

Interface(s) on which to listen to DHCPv4 queries. If omitted, DHCP relay will listen on all broadcast interfaces.

```
vrouter running config# vrf <vrf> dhcp relay dhcp-server <dhcp-server>
vrouter running dhcp-server <dhcp-server># interface INTERFACE
```

|           |                    |
|-----------|--------------------|
| INTERFACE | An interface name. |
|-----------|--------------------|

## handle-option

Handling of DHCPv4 packets that already contain relay agent options. Override the matching option in root context.

```
vrouter running config# vrf <vrf> dhcp relay dhcp-server <dhcp-server>
vrouter running dhcp-server <dhcp-server># handle-option HANDLE-OPTION
```

| HANDLE-OPTION values | Description  |
|----------------------|--|
| append               | Append our own set of relay options to the packet, leaving the supplied option field intact. |
| replace              | Replace the existing agent option field.   |
| forward              | Forward the packet unchanged.  |
| discard              | Discard the packet.  |

### drop-unmatched

If true, drop packets from upstream servers if they were generated in response to a different relay agent. Override the matching option in root context.

```
vrouter running config# vrf <vrf> dhcp relay dhcp-server <dhcp-server>  
vrouter running dhcp-server <dhcp-server># drop-unmatched true|false
```

### hop-count

Maximum hop count before packets are discarded. Override the matching option in root context.

```
vrouter running config# vrf <vrf> dhcp relay dhcp-server <dhcp-server>  
vrouter running dhcp-server <dhcp-server># hop-count <0-255>
```

### max-size

Maximum packet size to send to a DHCPv4 server. If a DHCP packet size surpasses this value it will be forwarded without appending relay agent information. Override the matching option in root context.

```
vrouter running config# vrf <vrf> dhcp relay dhcp-server <dhcp-server>  
vrouter running dhcp-server <dhcp-server># max-size <64-1400>
```

## 3.2.30 fast-path

---

**Note:** requires a Turbo Router Network License.

---

Fast path configuration.

```
vrouter running config# system fast-path
```

### enabled

Enable or disable the fast path.

```
vrouter running config# system fast-path  
vrouter running fast-path# enabled true|false
```

### Default value

true

## port

A physical network port managed by the fast path.

```
vrouter running config# system fast-path
vrouter running fast-path# port PORT
```

| PORT values        | Description            |
|--------------------|------------------------|
| <pci-port>         | PCI port name.         |
| <device-tree-port> | Device tree port name. |
| <device-tree-port> | Hyper-V port name.     |

## core-mask

Dedicate cores to fast path or exception path.

```
vrouter running config# system fast-path core-mask
```

## fast-path

List of cores dedicated to fast path.

```
vrouter running config# system fast-path core-mask
vrouter running core-mask# fast-path FAST-PATH
```

| FAST-PATH values | Description   |
|------------------|---|
| max              | Dedicate the maximum number of cores to the fast path.                  |
| half             | Dedicate half of the cores to the fast path.                            |
| min              | Dedicate the minimum number of cores to the fast path.                  |
| <cores-list>     | A comma-separated list of cores or core ranges. Example: '1,4-7,10-12'. |

## exception

Control plane cores allocated to exception packets processing. If unset, use the first non fast path core.

```
vrouter running config# system fast-path core-mask
vrouter running core-mask# exception EXCEPTION
```

|           |   |
|-----------|---|
| EXCEPTION | A comma-separated list of cores or core ranges. Example: '1,4-7,10-12'. |
|-----------|---|

## linux-to-fp

Fast path cores that can receive packets from Linux. It must be included in fast path mask. If unset, all fast path cores can receive packets from Linux.

```
vrouter running config# system fast-path core-mask
vrouter running core-mask# linux-to-fp LINUX-TO-FP
```

|             |   |
|-------------|---|
| LINUX-TO-FP | A comma-separated list of cores or core ranges. Example: '1,4-7,10-12'. |
|-------------|---|

## qos

Fast path cores dedicated for qos schedulers. These cores do not received any packets from the NIC or Linux.

```
vrouter running config# system fast-path core-mask
vrouter running core-mask# qos QOS
```

|     |   |
|-----|---|
| QOS | A comma-separated list of cores or core ranges. Example: '1,4-7,10-12'. |
|-----|---|

## port

Map fast path cores with network ports, specifying which logical cores poll which ports. Example: 'c1=0:1/c2=2/c3=0:1:2' means the logical core 1 polls the port 0 and 1, the core 2 polls the port 2, and the core 3 polls the ports 0, 1, and 2. If unset, each port is polled by all the logical cores of the same socket.

```
vrouter running config# system fast-path core-mask
vrouter running core-mask# port <core-port-map>
```

## cp-protection

Control plane protection configuration.

```
vrouter running config# system fast-path cp-protection
```

## budget

Maximum CPU usage allowed for Control Plane Protection in percent.

```
vrouter running config# system fast-path cp-protection
vrouter running cp-protection# budget <int16>
```

### Default value

10

## crypto

Fast path crypto configuration.

```
vrouter running config# system fast-path crypto
```

## driver

Crypto driver. If unset, select automatically.

```
vrouter running config# system fast-path crypto
vrouter running crypto# driver DRIVER
```

| DRIVER values | Description                |
|---------------|----------------------------|
| multibuffer   | Intel multibuffer library. |
| quickassist   | Intel quickassist.         |
| dppk-pmd      | DPDK crypto PMD.           |
| octeontxcpt   | Marvell Octeon TX.         |
| octeontx2cpt  | Marvell Octeon TX2.        |

## offload-core-mask

Fast path cores that can do crypto operations for other fast path cores. It must be included in fast path mask. The crypto offloading is always done on cores in the same NUMA node.

```
vrouter running config# system fast-path crypto
vrouter running crypto# offload-core-mask OFFLOAD-CORE-MASK
```

| OFFLOAD-CORE-MASK values | Description   |
|--------------------------|---|
| <cores-list>             | A comma-separated list of cores or core ranges. Example: '1,4-7,10-12'. |
| none                     | Disable crypto offload.   |

### nb-session

Maximum number of cryptographic sessions.

```
vrouter running config# system fast-path crypto
vrouter running crypto# nb-session <uint32>
```

### nb-buffer

Maximum number of cryptographic buffers, representing the maximum number of in-flight operations, either being processed by the asynchronous crypto engine, or waiting in crypto device queues.

```
vrouter running config# system fast-path crypto
vrouter running crypto# nb-buffer <uint32>
```

### advanced

Advanced configuration for fast path.

```
vrouter running config# system fast-path advanced
```

### nb-mbuf

Number of mbufs (network packet descriptors). The value can be an integer representing the total number of mbufs, an integer prefixed with '+' representing the number of mbufs to add to the automatic value. In case of NUMA, the value can be a per-socket list. If unset, nb-mbuf is determined automatically.

```
vrouter running config# system fast-path advanced
vrouter running advanced# nb-mbuf <nb-mbuf>
```

### machine-memory

Set the memory that will be used by the fast path (hugepages, shm, mallocs...) so it can run on a machine with this amount of physical memory.

```
vrouter running config# system fast-path advanced
vrouter running advanced# machine-memory <uint32>
```

### mainloop-sleep-delay

If set, add a sleep time after each idle mainloop turn. This will drastically decrease performance.

```
vrouter running config# system fast-path advanced
vrouter running advanced# mainloop-sleep-delay <uint16>
```

### offload

Enable or disabled advanced offload features such as TSO, L4 checksum offloading, or offload information forwarding from a guest to the NIC through a virtual interface. If unset, use default product configuration.

```
vrouter running config# system fast-path advanced
vrouter running advanced# offload true|false
```

### vlan-strip

Strip the VLAN header from incoming frames if supported by the hardware. By default, vlan stripping feature is disabled.

```
vrouter running config# system fast-path advanced
vrouter running advanced# vlan-strip true|false
```

### intercore-ring-size

Set the size of the intercore rings, used by dataplane cores to send messages to another dataplane core. The default size depends on the product.

```
vrouter running config# system fast-path advanced
vrouter running advanced# intercore-ring-size <uint16>
```

### software-txq

Set the default size of Tx software queue. This field must be a power of 2. Default is 0 (no software queue).

```
vrouter running config# system fast-path advanced
vrouter running advanced# software-txq <uint16>
```

### nb-rxd

Set the default number of Rx hardware descriptors for Ethernet ports. The value must be accepted by all devices on the system. If unset, an automatic value is used.

```
vrouter running config# system fast-path advanced
vrouter running advanced# nb-rxd <uint16>
```

### nb-txd

Set the default number of Tx hardware descriptors for Ethernet ports. The value must be accepted by all devices on the system. If unset, an automatic value is used.

```
vrouter running config# system fast-path advanced
vrouter running advanced# nb-txd <uint16>
```

### reserve-hugepages

Enable or disable the automatic huge pages allocation by the fast path. When disabled, the user is responsible for providing enough huge pages for the fast path to start.

```
vrouter running config# system fast-path advanced
vrouter running advanced# reserve-hugepages true|false
```

### ipv4-netfilter-cache

Enable or disable the IPv4 netfilter cache.

```
vrouter running config# system fast-path advanced
vrouter running advanced# ipv4-netfilter-cache true|false
```

#### Default value

true

## ipv6-netfilter-cache

Enable or disable the IPv6 netfilter cache.

```
vrouter running config# system fast-path advanced
vrouter running advanced# ipv6-netfilter-cache true|false
```

### Default value

true

## ipv4-pre-ipsec-fragmentation

Configure IPv4 pre IPsec fragmentation. When enabled, this behavior helps releasing pressure on the decrypting device, as the reassembly will be done on the destination host of the inner packet instead of the decrypting device. It applies only in tunnel mode.

```
vrouter running config# system fast-path advanced
vrouter running advanced# ipv4-pre-ipsec-fragmentation IPV4-PRE-IPSEC-FRAGMENTATION
```

| IPV4-PRE-IPSEC-FRAGMENTATION values | Description   |
|-------------------------------------|---|
| always                              | Pre IPsec fragmentation is always performed.  |
| check-df-bit                        | Pre IPsec fragmentation is performed only if the don't fragment bit is not set on the inner packet. Applies only to IPv4 inner packets. |
| off                                 | Post IPsec fragmentation is performed.  |

### Default value

off

## ipv6-pre-ipsec-fragmentation

Configure IPv6 pre IPsec fragmentation. When enabled, this behavior helps releasing pressure on the decrypting device, as the reassembly will be done on the destination host of the inner packet instead of the decrypting device. It applies only in tunnel mode.

```
vrouter running config# system fast-path advanced
vrouter running advanced# ipv6-pre-ipsec-fragmentation IPV6-PRE-IPSEC-FRAGMENTATION
```

| IPV6-PRE-IPSEC-FRAGMENTATION values | Description   |
|-------------------------------------|---|
| always                              | Pre IPsec fragmentation is always performed.  |
| check-df-bit                        | Pre IPsec fragmentation is performed only if the don't fragment bit is not set on the inner packet. Applies only to IPv4 inner packets. |
| off                                 | Post IPsec fragmentation is performed.  |

### Default value

off

### hardware-queue-map

Hardware queue map used to change the destination queue according the hash computed on the packet from the RSS function.

```
vrouter running config# system fast-path advanced
vrouter running advanced# hardware-queue-map <port> <uint16> <uint16>
```

|        |                |
|--------|----------------|
| <port> | PCI port name. |
|--------|----------------|

|          |                                 |
|----------|---------------------------------|
| <uint16> | Hardware queue map table index. |
|----------|---------------------------------|

|          |                       |
|----------|-----------------------|
| <uint16> | Destination Rx queue. |
|----------|-----------------------|

### limits

Global runtime limits for fast path.

```
vrouter running config# system fast-path limits
```

### fp-max-if

Maximum number of interfaces. It includes physical ports and virtual interfaces like gre, vlan, ...

```
vrouter running config# system fast-path limits
vrouter running limits# fp-max-if <uint32>
```

### fp-max-vrf

Maximum number of VRFs.

```
vrouter running config# system fast-path limits
vrouter running limits# fp-max-vrf <uint32>
```

### ip4-max-addr

Maximum number of IPv4 addresses.

```
vrouter running config# system fast-path limits
vrouter running limits# ip4-max-addr <uint32>
```

### ip4-max-route

Maximum number of IPv4 routes.

```
vrouter running config# system fast-path limits
vrouter running limits# ip4-max-route <uint32>
```

### ip4-max-neigh

Maximum number of IPv4 neighbors.

```
vrouter running config# system fast-path limits
vrouter running limits# ip4-max-neigh <uint32>
```

### ip6-max-addr

Maximum number of IPv6 addresses.

```
vrouter running config# system fast-path limits
vrouter running limits# ip6-max-addr <uint32>
```

### ip6-max-route

Maximum number of IPv6 routes.

```
vrouter running config# system fast-path limits
vrouter running limits# ip6-max-route <uint32>
```

### ip6-max-neigh

Maximum number of IPv6 neighbors.

```
vrouter running config# system fast-path limits
vrouter running limits# ip6-max-neigh <uint32>
```

### pbr-max-rule

Maximum number of PBR rules.

```
vrouter running config# system fast-path limits
vrouter running limits# pbr-max-rule <uint32>
```

### filter4-max-rule

Maximum number of IPv4 Netfilter rules.

```
vrouter running config# system fast-path limits
vrouter running limits# filter4-max-rule <uint32>
```

### filter6-max-rule

Maximum number of IPv6 Netfilter rules.

```
vrouter running config# system fast-path limits
vrouter running limits# filter6-max-rule <uint32>
```

### filter4-max-ct

Maximum number of IPv4 Netfilter conntracks.

```
vrouter running config# system fast-path limits  
vrouter running limits# filter4-max-ct <uint32>
```

### filter6-max-ct

Maximum number of IPv6 Netfilter conntracks.

```
vrouter running config# system fast-path limits  
vrouter running limits# filter6-max-ct <uint32>
```

### filter-max-ipset

Maximum number of ipsets per VRF.

```
vrouter running config# system fast-path limits  
vrouter running limits# filter-max-ipset <uint32>
```

### filter-max-ipset-entry

Maximum number of entries per ipset.

```
vrouter running config# system fast-path limits  
vrouter running limits# filter-max-ipset-entry <uint32>
```

### filter-bridge-max-rule

Maximum number of bridge filter rules.

```
vrouter running config# system fast-path limits  
vrouter running limits# filter-bridge-max-rule <uint32>
```

### vxlan-max-port

Maximum number of (VXLAN destination port, VRF) pairs.

```
vrouter running config# system fast-path limits
vrouter running limits# vxlan-max-port <uint32>
```

### vxlan-max-if

Maximum number of VXLAN interfaces.

```
vrouter running config# system fast-path limits
vrouter running limits# vxlan-max-if <uint32>
```

### vxlan-max-fdb

Maximum number of VXLAN forwarding database entries.

```
vrouter running config# system fast-path limits
vrouter running limits# vxlan-max-fdb <uint32>
```

### reass4-max-queue

Maximum number of simultaneous reassembly procedures for IPv4.

```
vrouter running config# system fast-path limits
vrouter running limits# reass4-max-queue <uint32>
```

### reass6-max-queue

Maximum number of simultaneous reassembly procedures for IPv6.

```
vrouter running config# system fast-path limits
vrouter running limits# reass6-max-queue <uint32>
```

### ipsec-max-sp

Maximum number of IPv4 and IPv6 IPsec SPs.

```
vrouter running config# system fast-path limits
vrouter running limits# ipsec-max-sp <uint32>
```

### ipsec-max-sa

Maximum number of IPv4 and IPv6 IPsec SAs.

```
vrouter running config# system fast-path limits
vrouter running limits# ipsec-max-sa <uint32>
```

### ip-max-8-table (deprecated)

**Attention:**

**Deprecated since:** 2021-04-12

**Obsolete in release:** 21q3

**Description:** This option is not relevant with new LPM algorithm.

**Replacement:** / system fast-path limits ip-max-lpm-memory

Maximum number of IPv4 and IPv6 /8 table entries.

```
vrouter running config# system fast-path limits
vrouter running limits# ip-max-8-table <uint32>
```

### ip-max-lpm-table

Maximum number of IPv4 and IPv6 tables.

```
vrouter running config# system fast-path limits
vrouter running limits# ip-max-lpm-table <uint32>
```

### ip-max-lpm-memory

Amount of memory reserved for IPv4 and IPv6 LPM tree.

```
vrouter running config# system fast-path limits  
vrouter running limits# ip-max-lpm-memory <uint32>
```

### filter-max-cache

Maximum number of IPv4 flows stored in filter cache.

```
vrouter running config# system fast-path limits  
vrouter running limits# filter-max-cache <uint32>
```

### filter6-max-cache

Maximum number of IPv6 flows stored in filter cache.

```
vrouter running config# system fast-path limits  
vrouter running limits# filter6-max-cache <uint32>
```

### vlan-max-if

Maximum number of VLAN interfaces.

```
vrouter running config# system fast-path limits  
vrouter running limits# vlan-max-if <uint32>
```

### macvlan-max-if

Maximum number of MACVLAN (VRRP) interfaces.

```
vrouter running config# system fast-path limits  
vrouter running limits# macvlan-max-if <uint32>
```

### gre-max-if

Maximum number of GRE interfaces.

```
vrouter running config# system fast-path limits
vrouter running limits# gre-max-if <uint32>
```

### svti-max-if

Maximum number of SVTI interfaces.

```
vrouter running config# system fast-path limits
vrouter running limits# svti-max-if <uint32>
```

### fp-cur-if (state only)

Current number of interfaces. It includes physical ports and virtual interfaces like gre, vlan, ...

```
vrouter> show state system fast-path limits fp-cur-if
```

### fp-cur-vrf (state only)

Current number of VRFs.

```
vrouter> show state system fast-path limits fp-cur-vrf
```

### ip4-cur-addr (state only)

Current number of IPv4 addresses.

```
vrouter> show state system fast-path limits ip4-cur-addr
```

### ip4-cur-route (state only)

Current number of IPv4 routes.

```
vrouter> show state system fast-path limits ip4-cur-route
```

### ip4-cur-neigh (state only)

Current number of IPv4 neighbors.

```
vrouter> show state system fast-path limits ip4-cur-neigh
```

### ip6-cur-addr (state only)

Current number of IPv6 addresses.

```
vrouter> show state system fast-path limits ip6-cur-addr
```

### ip6-cur-route (state only)

Current number of IPv6 routes.

```
vrouter> show state system fast-path limits ip6-cur-route
```

### ip6-cur-neigh (state only)

Current number of IPv6 neighbors.

```
vrouter> show state system fast-path limits ip6-cur-neigh
```

### pbr-cur-rule (state only)

Current number of PBR rules.

```
vrouter> show state system fast-path limits pbr-cur-rule
```

### filter4-cur-rule (state only)

Current number of IPv4 Netfilter rules.

```
vrouter> show state system fast-path limits filter4-cur-rule
```

**filter6-cur-rule (state only)**

Current number of IPv6 Netfilter rules.

```
vrouter> show state system fast-path limits filter6-cur-rule
```

**filter4-cur-ct (state only)**

Current number of IPv4 Netfilter conntracks.

```
vrouter> show state system fast-path limits filter4-cur-ct
```

**filter6-cur-ct (state only)**

Current number of IPv6 Netfilter conntracks.

```
vrouter> show state system fast-path limits filter6-cur-ct
```

**filter-cur-ipset (state only)**

Current number of ipsets per VRF.

```
vrouter> show state system fast-path limits filter-cur-ipset
```

**vxlan-cur-port (state only)**

Current number of (VXLAN destination port, VRF) pairs.

```
vrouter> show state system fast-path limits vxlan-cur-port
```

**vxlan-cur-if (state only)**

Current number of VXLAN interfaces.

```
vrouter> show state system fast-path limits vxlan-cur-if
```

### vxlan-cur-fdb (state only)

Current number of VXLAN forwarding database entries.

```
vrouter> show state system fast-path limits vxlan-cur-fdb
```

### ipsec-cur-sp (state only)

Current number of IPv4 and IPv6 IPsec SPs.

```
vrouter> show state system fast-path limits ipsec-cur-sp
```

### ipsec-cur-sa (state only)

Current number of IPv4 and IPv6 IPsec SAs.

```
vrouter> show state system fast-path limits ipsec-cur-sa
```

### ip-cur-8-table (deprecated) (state only)

**Attention:**

**Deprecated since:** 2021-04-12

**Obsolete in release:** 21q3

**Description:** This option is not relevant with new LPM algorithm. It is replaced by ip-max-lpm-memory, which can be set to (ip-max-8-table / 512).

**Replacement:** state system fast-path limits ip-cur-lpm-memory

Current number of IPv4 and IPv6 /8 table entries.

```
vrouter> show state system fast-path limits ip-cur-8-table
```

### ip-cur-lpm-table (state only)

Current number of IPv4 and IPv6 tables.

```
vrouter> show state system fast-path limits ip-cur-lpm-table
```

### ip-cur-lpm-memory (state only)

Current amount of memory reserved for IPv4 and IPv6 LPM tree.

```
vrouter> show state system fast-path limits ip-cur-lpm-memory
```

### vlan-cur-if (state only)

Current number of VLAN interfaces.

```
vrouter> show state system fast-path limits vlan-cur-if
```

### macvlan-cur-if (state only)

Current number of MACVLAN (VRRP) interfaces.

```
vrouter> show state system fast-path limits macvlan-cur-if
```

### gre-cur-if (state only)

Current number of GRE interfaces.

```
vrouter> show state system fast-path limits gre-cur-if
```

### svti-cur-if (state only)

Current number of SVTI interfaces.

```
vrouter> show state system fast-path limits svti-cur-if
```

## cg-nat

Fast path cg-nat configuration.

```
vrouter running config# system fast-path limits cg-nat
```

## max-contracks

Maximum number of tracked connections.

```
vrouter running config# system fast-path limits cg-nat  
vrouter running cg-nat# max-contracks <uint32>
```

## max-nat-entries

Maximum number of NAT translations.

```
vrouter running config# system fast-path limits cg-nat  
vrouter running cg-nat# max-nat-entries <uint32>
```

## max-users

Maximum number of users.

```
vrouter running config# system fast-path limits cg-nat  
vrouter running cg-nat# max-users <uint32>
```

## max-blocks

Maximum number of blocks.

```
vrouter running config# system fast-path limits cg-nat  
vrouter running cg-nat# max-blocks <uint32>
```

### max-block-size

Maximum number of ports per block.

```
vrouter running config# system fast-path limits cg-nat
vrouter running cg-nat# max-block-size <uint32>
```

### linux-sync

Advanced tuning for fast path / Linux synchronization.

```
vrouter running config# system fast-path linux-sync
```

### fpm-socket-size

Buffer size of the socket used to communicate between the cache manager and the fast path manager.

```
vrouter running config# system fast-path linux-sync
vrouter running linux-sync# fpm-socket-size <uint32>
```

#### Default value

2097152

### nl-socket-size

Buffer size of the cache manager netlink socket.

```
vrouter running config# system fast-path linux-sync
vrouter running linux-sync# nl-socket-size <uint32>
```

#### Default value

67108864

### ipset-dump-delay

Delay period for polling the ipset content.

```
vrouter running config# system fast-path linux-sync
vrouter running linux-sync# ipset-dump-delay <uint32>
```

#### Default value

1

**disable**

Disable synchronization for specific modules.

```
vrouter running config# system fast-path linux-sync
vrouter running linux-sync# disable DISABLE
```

| DISABLE values | Description  |
|----------------|--|
| bpf            | Disable BPF synchronization (used by traffic capture).                             |
| bridge         | Disable bridge interface synchronization.  |
| conntrack      | Disable connection tracking synchronization.                                       |
| firewall       | Disable firewall synchronization.  |
| gre            | Disable GRE interface synchronization.   |
| ipip           | Disable IP in IP interface synchronization.  |
| ipsec          | Disable IPsec synchronization.   |
| ipset4         | Disable IPv4 ipset synchronization (used by firewall IPv4 address/network groups). |
| ipset6         | Disable IPv6 ipset synchronization (used by firewall IPv6 address/network groups). |
| ipv6           | Disable IPv6 synchronization.  |
| lag            | Disable LAG interface synchronization.   |
| macvlan        | Disable MACVLAN interface synchronization (used by VRRP).                          |
| mpls           | Disable MPLS synchronization.  |
| nat            | Disable NAT synchronization.   |
| svti           | Disable SVTI interface synchronization.  |
| vlan           | Disable VLAN interface synchronization.  |
| vxlan          | Disable VXLAN interface synchronization.   |

**cpu-usage (state only)**

The list of busy percentage per CPU.

**busy (state only)**

The busy percentage.

```
vrouter> show state system fast-path cpu-usage <string> busy
```

### 3.2.31 logging

#### Global Settings

---

**Note:** requires a Turbo Router Network License.

---

Global logging configuration.

```
vrouter running config# system logging
```

#### disk-usage (state only)

Total disk usage of all journal files.

```
vrouter> show state system logging disk-usage
```

#### rate-limit

Configure logging rate limiting.

```
vrouter running config# system logging rate-limit
```

#### interval

Amount of time that is being measured for rate limiting. A value of 0 disables rate limiting.

```
vrouter running config# system logging rate-limit  
vrouter running rate-limit# interval <uint32>
```

#### Default value

30

## burst

Amount of messages that have to occur in the rate limit interval to trigger rate limiting. A value of 0 disables rate limiting.

```
vrouter running config# system logging rate-limit
vrouter running rate-limit# burst <uint32>
```

### Default value

1000

## Per-VRF Settings

---

**Note:** requires a Turbo Router Network License.

---

Per-VRF logging configuration.

```
vrouter running config# vrf <vrf> logging
```

## syslog

Syslog configuration.

```
vrouter running config# vrf <vrf> logging syslog
```

## enabled

Enable syslog.

```
vrouter running config# vrf <vrf> logging syslog
vrouter running syslog# enabled true|false
```

### Default value

true

### remote-server

Remote log server list.

```
vrouter running config# vrf <vrf> logging syslog remote-server <remote-server>
```

| <remote-server> values | Description  |
|------------------------|--|
| <A.B.C.D>              | IPv4 address.  |
| <X:X::X:X>             | IPv6 address.  |
| <host-name>            | The domain-name type represents a DNS domain name. Fully qualified left to the models which utilize this type. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be encoded in punycode as described in RFC 3492. |

### protocol

Transmission protocol.

```
vrouter running config# vrf <vrf> logging syslog remote-server <remote-server>
vrouter running remote-server <remote-server># protocol PROTOCOL
```

| PROTOCOL values | Description   |
|-----------------|---|
| udp             | Traditional UDP transport. Extremely lossy but standard.  |
| tcp             | Plain TCP based transport. Loses messages only during certain situations but is widely available. |

### Default value

tcp

## port

Sets the destination port number for syslog UDP messages to the server.

```
vrouter running config# vrf <vrf> logging syslog remote-server <remote-server>
vrouter running remote-server <remote-server># port PORT
```

|      |   |
|------|---|
| PORT | A 16-bit port number used by a transport protocol such as TCP or UDP. |
|------|---|

### Default value

514

## log-filter

Filter messages sent to the server.

```
vrouter running config# vrf <vrf> logging syslog remote-server <remote-server>
vrouter running remote-server <remote-server># log-filter facility <log-filter> \
... level EQUAL greater-or-equal GREATER-OR-EQUAL \
... not LEVEL
```

| <log-filter> values | Description  |
|---------------------|--|
| kernel              | Filter kernel messages.                                    |
| mail                | Filter mail system messages.                               |
| news                | Filter network news subsystem messages.                    |
| user                | Filter random user-level messages.                         |
| auth                | Filter security/authorization messages.                    |
| authpriv            | Filter security/authorization messages (private).          |
| cron                | Filter clock daemon messages.                              |
| daemon              | Filter system daemons messages.                            |
| line-printer        | Filter line printer subsystem messages.                    |
| FTP                 | Filter FTP daemon messages.                                |
| syslog              | Filter messages generated internally by the syslog daemon. |
| uucp                | Filter UUCP subsystem messages.                            |
| local0              | Filter messages from local0.                               |
| local1              | Filter messages from local1.                               |
| local2              | Filter messages from local2.                               |
| local3              | Filter messages from local3.                               |
| local4              | Filter messages from local4.                               |
| local5              | Filter messages from local5.                               |
| local6              | Filter messages from local6.                               |
| local7              | Filter messages from local7.                               |
| any                 | Filter messages from any facilities.                       |

### level

Select messages level to send to the server.

```
level EQUAL greater-or-equal GREATER-OR-EQUAL \
not LEVEL
```

### EQUAL

Select levels to send the server.

```
EQUAL
```

| EQUAL values | Description                           |
|--------------|---------------------------------------|
| emergency    | System is unusable.                   |
| alert        | Action must be taken immediately.     |
| critical     | Critical conditions.                  |
| error        | Error conditions.                     |
| warning      | Warning conditions.                   |
| notice       | Normal but significant condition.     |
| info         | Informational messages.               |
| debug        | Debug-level messages.                 |
| any          | Send all messages from this facility. |
| none         | Send nothing from this facility.      |

### greater-or-equal

Send messages with a greater or equal level than the selected one to the server.

```
greater-or-equal GREATER-OR-EQUAL
```

| GREATER-OR-EQUAL values | Description                       |
|-------------------------|-----------------------------------|
| emergency               | System is unusable.               |
| alert                   | Action must be taken immediately. |
| critical                | Critical conditions.              |
| error                   | Error conditions.                 |
| warning                 | Warning conditions.               |
| notice                  | Normal but significant condition. |
| info                    | Informational messages.           |
| debug                   | Debug-level messages.             |

**not**

Select levels to not send to the server.

```
not LEVEL
```

**LEVEL**

Do not send messages with this level.

```
LEVEL
```

| LEVEL values | Description                       |
|--------------|-----------------------------------|
| emergency    | System is unusable.               |
| alert        | Action must be taken immediately. |
| critical     | Critical conditions.              |
| error        | Error conditions.                 |
| warning      | Warning conditions.               |
| notice       | Normal but significant condition. |
| info         | Informational messages.           |
| debug        | Debug-level messages.             |

**tls**

Enable syslog messages encryption and server/client authentication.

```
vrouter running config# vrf <vrf> logging syslog tls
```

**enabled**

Enable/disable syslog messages encryption and server/client authentication.

```
vrouter running config# vrf <vrf> logging syslog tls
vrouter running tls# enabled true|false
```

**Default value**

true

### ca-certificate (mandatory)

PEM-encoded X509 certificate authority certificate.

```
vrouter running config# vrf <vrf> logging syslog tls
vrouter running tls# ca-certificate <string>
```

### certificate

PEM-encoded X509 certificate.

```
vrouter running config# vrf <vrf> logging syslog tls
vrouter running tls# certificate <string>
```

### private-key

PEM-encoded X509 private key.

```
vrouter running config# vrf <vrf> logging syslog tls
vrouter running tls# private-key <string>
```

### server-authentication

Server authentication mode selection.

```
vrouter running config# vrf <vrf> logging syslog tls
vrouter running tls# server-authentication anonymous certificate \
... name <string> \
... fingerprint <string>
```

### anonymous

No authentication.

```
anonymous
```

**certificate**

Certificate validation only.

```
certificate
```

**name**

Certificate validation and subject name authentication.

```
name <string>
```

**<string>**

Certificate validation and subject name authentication.

```
<string>
```

**fingerprint**

Certificate fingerprint authentication.

```
fingerprint <string>
```

**<string>**

Certificate fingerprint authentication.

```
<string>
```

**3.2.32 high availability****HA groups**

---

**Note:** requires a Turbo Router Network License.

---

Global high-availability configuration.

```
vrouter running config# ha
```

## group

The list of high-availability groups on the device, used to advertise an high-availability status. Each group can be associated to one notifier and several subscribers.

```
vrouter running config# ha group <group>
```

|         |                             |
|---------|-----------------------------|
| <group> | An high-availability group. |
|---------|-----------------------------|

## state

Force the high-availability state of this group.

```
vrouter running config# ha group <group>
vrouter running group <group># state STATE
```

| STATE values | Description       |
|--------------|-------------------|
| master       | Set master state. |
| backup       | Set backup state. |

## HA neighbor

---

**Note:** requires a Turbo Router Network License.

---

High-availability neighbor configuration.

```
vrouter running config# vrf <vrf> ha-neighbor
```

## enabled

Enable/Disable HA neighbor in this VRF.

```
vrouter running config# vrf <vrf> ha-neighbor
vrouter running ha-neighbor# enabled true|false
```

## Default value

true

### node-id (mandatory)

The local node ID.

```
vrouter running config# vrf <vrf> ha-neighbor
vrouter running ha-neighbor# node-id <uint8>
```

### local-address (mandatory)

The local IP address used to accept remote peer connections.

```
vrouter running config# vrf <vrf> ha-neighbor
vrouter running ha-neighbor# local-address LOCAL-ADDRESS
```

| LOCAL-ADDRESS values | Description      |
|----------------------|------------------|
| <A.B.C.D>            | An IPv4 address. |
| <X:X::X:X>           | An IPv6 address. |

### listen-ha-group (mandatory)

The HA group to be monitored.

```
vrouter running config# vrf <vrf> ha-neighbor
vrouter running ha-neighbor# listen-ha-group <string>
```

### interface (mandatory)

Interface used for peer discovery in multicast mode.

```
vrouter running config# vrf <vrf> ha-neighbor
vrouter running ha-neighbor# interface INTERFACE
```

|           |                    |
|-----------|--------------------|
| INTERFACE | An interface name. |
|-----------|--------------------|

**state (state only)**

The high-availability state.

```
vrouter> show state vrf <vrf> ha-neighbor state
```

**HA contrack**


---

**Note:** requires a Turbo Router Network License.

---

High-availability contrack configuration.

```
vrouter running config# vrf <vrf> ha-contrack
```

**enabled**

Enable/Disable HA contrack in this VRF.

```
vrouter running config# vrf <vrf> ha-contrack
vrouter running ha-contrack# enabled true|false
```

**Default value**

true

**local-address (mandatory)**

The local IP address used to accept remote peer connections.

```
vrouter running config# vrf <vrf> ha-contrack
vrouter running ha-contrack# local-address LOCAL-ADDRESS
```

| LOCAL-ADDRESS values | Description      |
|----------------------|------------------|
| <A.B.C.D>            | An IPv4 address. |
| <X:X::X:X>           | An IPv6 address. |

### listen-ha-group (mandatory)

The HA group to be monitored.

```
vrouter running config# vrf <vrf> ha-contrack  
vrouter running ha-contrack# listen-ha-group <string>
```

### interface (mandatory)

Interface used to send synchronization messages.

```
vrouter running config# vrf <vrf> ha-contrack  
vrouter running ha-contrack# interface INTERFACE
```

|           |                    |
|-----------|--------------------|
| INTERFACE | An interface name. |
|-----------|--------------------|

### state (state only)

The high-availability contrack state.

```
vrouter> show state vrf <vrf> ha-contrack state
```

### protocol-list

Configure protocols to accept or ignore.

```
vrouter running config# vrf <vrf> ha-contrack protocol-list
```

### accept (mandatory)

Accept or ignore protocols.

```
vrouter running config# vrf <vrf> ha-contrack protocol-list  
vrouter running protocol-list# accept true|false
```

## protocol

Protocol list to accept or ignore.

```
vrouter running config# vrf <vrf> ha-contrack protocol-list
vrouter running protocol-list# protocol PROTOCOL
```

| PROTOCOL values | Description                |
|-----------------|----------------------------|
| tcp             | Add TCP in the list.       |
| sctp            | Add SCTP in the list.      |
| dccp            | Add DCCP in the list.      |
| udp             | Add UDP in the list.       |
| icmp            | Add ICMP in the list.      |
| ipv6-icmp       | Add IPv6 ICMP in the list. |

## address-list

Configure addresses to accept or ignore.

```
vrouter running config# vrf <vrf> ha-contrack address-list
```

### accept (mandatory)

Accept or ignore addresses.

```
vrouter running config# vrf <vrf> ha-contrack address-list
vrouter running address-list# accept true|false
```

## address

Address list to accept or ignore.

```
vrouter running config# vrf <vrf> ha-contrack address-list
vrouter running address-list# address ADDRESS
```

| ADDRESS values | Description      |
|----------------|------------------|
| <A.B.C.D>      | An IPv4 address. |
| <X:X::X:X>     | An IPv6 address. |

### 3.2.33 group

Address and network group configuration. They can then be used in the firewall configuration.

```
vrouter running config# vrf <vrf> group
```

#### ipv4

IPv4 address and network group configuration.

```
vrouter running config# vrf <vrf> group ipv4
```

#### address-group

Address group.

```
vrouter running config# vrf <vrf> group ipv4 address-group <string>
```

|          |                            |
|----------|----------------------------|
| <string> | Name of the address group. |
|----------|----------------------------|

#### address

List of addresses of the group.

```
vrouter running config# vrf <vrf> group ipv4 address-group <string>
vrouter running address-group <string># address ADDRESS
```

|              |   |
|--------------|---|
| AD-<br>DRESS | An IPv4 address without a zone index. This type, derived from ipv4-address, may be used in situations where the zone is known from the context and hence no zone index is needed. |
|--------------|---|

#### used (state only)

The address-group is in use.

```
vrouter> show state vrf <vrf> group ipv4 address-group <string> used
```

## network-group

Network group.

```
vrouter running config# vrf <vrf> group ipv4 network-group <string>
```

|          |                            |
|----------|----------------------------|
| <string> | Name of the network group. |
|----------|----------------------------|

## network

List of networks of the group.

```
vrouter running config# vrf <vrf> group ipv4 network-group <string>  
vrouter running network-group <string># network NETWORK
```

|         |  |
|---------|--|
| NETWORK | An IPv4 prefix: address and CIDR mask. |
|---------|--|

## used (state only)

The network-group is in use.

```
vrouter> show state vrf <vrf> group ipv4 network-group <string> used
```

## ipv6

IPv6 address and network group configuration.

```
vrouter running config# vrf <vrf> group ipv6
```

## address-group

Address group.

```
vrouter running config# vrf <vrf> group ipv6 address-group <string>
```

|          |                            |
|----------|----------------------------|
| <string> | Name of the address group. |
|----------|----------------------------|

## address

List of addresses of the group.

```
vrouter running config# vrf <vrf> group ipv6 address-group <string>
vrouter running address-group <string># address ADDRESS
```

|              |   |
|--------------|---|
| AD-<br>DRESS | An IPv6 address without a zone index. This type, derived from ipv6-address, may be used in situations where the zone is known from the context and hence no zone index is needed. |
|--------------|---|

## used (state only)

The address-group is in use.

```
vrouter> show state vrf <vrf> group ipv6 address-group <string> used
```

## network-group

Network group.

```
vrouter running config# vrf <vrf> group ipv6 network-group <string>
```

|          |                            |
|----------|----------------------------|
| <string> | Name of the network group. |
|----------|----------------------------|

## network

List of networks of the group.

```
vrouter running config# vrf <vrf> group ipv6 network-group <string>
vrouter running network-group <string># network NETWORK
```

|         |  |
|---------|--|
| NETWORK | An IPv6 prefix: address and CIDR mask. |
|---------|--|

**used (state only)**

The network-group is in use.

```
vrouter> show state vrf <vrf> group ipv6 network-group <string> used
```

## 4. Troubleshooting

This guide references common configuration issues one may encounter when using Turbo Router, and indications on how to address them. These indications suppose you are logged as root and have access to the Linux shell.

### 4.1 Relevant Information for Bug Reporting

In case you cannot investigate and resolve the issue by yourself using this document, make sure you open a ticket on your 6WIND Customer Zone with the relevant troubleshooting information.

This information can be generated and exported using the following commands:

```
vrouter> cmd troubleshooting-report new
Gathering information. This may take some time...
Saved into /var/lib/yams/troubleshooting-reports/2018-09-24_17-27-07.tgz
vrouter> cmd troubleshooting-report export 2018-09-24_17-27-07.tgz url scp://
↪ john:s3cr3t@10.1.2.3/home/john
OK.
vrouter>
```

#### See also:

The *CLI User Guide*, Basics / Commands section for details.

The troubleshooting report includes the following information:

- Linux networking information
  - Stats on all known links
  - interfaces, addresses, routes, neighbours and IPsec
  - active network connections
  - Netfilter tables, bridge
- system information
  - topology
  - processors hierarchy
  - interrupts
  - memory

- PCI peripherals
- DMI/MBIOS
- kernel version, logs, cmdline and loaded modules
- distribution
- services list
- logs
- processes list
- cpuset
- devices (/dev)
- IRQ affinity
- mounted partitions
- core dumps
- fast path information
  - configuration, version, logs, status
  - debug info (ports, tables, etc.)
- running and startup configurations
- license information
  - average network throughput (measured in Rx)
  - average and current number of IPsec tunnels
  - average and current number of CG-NAT connections

## 4.2 Typical issues

### 4.2.1 Startup Issues

#### Turbo Router cannot start

##### Symptoms

- `systemctl status turbo` shows issues

##### Hints

- On Intel and Arm, check whether the configuration file is correct by looking at `fast-path.sh config` output for relevancy, and by checking config file syntactic correctness with `fast-path.sh config -c`.

Follow the advice regarding deprecated options as it may become problematic in later versions. Take into account the WARNINGS in the output.

- If you tried running the fast path and it crashed or failed along the way, some “runtime-only” files may be left unremoved. Make sure to call `fast-path.sh stop` before trying to start the fast path again.
- Look for error messages either on the console or in the logs. See *rsyslog* and *journalctl* sections for details regarding what can be found in the logs.
- Executable paths may change between two Turbo Router versions. Some shells (bash for example) keep a cache of the executable paths. After upgrading Turbo Router, if some commands are not found, you may need to start a new shell.

## Hugepages fragmentation

### Symptoms

- One of the following messages appears on the console or in the logs:

```
No more huge pages left for fastpath initialization

EAL: Not enough memory available! Requested: <X>MB, available: <Y smaller_
↳than X>MB
PANIC in rte_eal_init(): Cannot init memory

EAL: rte_eal_common_log_init(): cannot create log_history mempool
PANIC in rte_eal_init():
Cannot init logs

Not enough physically contiguous memory to allocate the mbuf pool on this_
↳socket (0): max_seg_size=178257920, total_mem=459276288, nb_seg=35
Increase the number of huge pages, use larger huge pages, or reboot the_
↳machine
PANIC in fpm_socket_mbufpool_create():
Cannot create mbuf pool for socket 0
```

### Hints

- There is a problem with the available memory.
- Add more memory.
- Check the output from `/proc/meminfo`, especially the `MemFree` and `HugePage_Free` fields. See *mem-info* section for details.

**MemFree** gives an indication of how much memory you may use for the fast path shared memory.

**HugePage\_Free** indicates how many huge pages are available for use by the fast path.

Beware, if hugepages are fragmented, you need to allocate more or simply reboot, as the DPDK requires contiguous physical memory.

## Not enough memory

### Symptoms

- The following message appears on the console or in the logs (and subsequent commands fail with similar messages):

```
/usr/bin/fast-path.sh: 435: /usr/bin/fast-path.sh: Cannot fork
/usr/bin/fast-path.sh: 668: /usr/bin/fast-path.sh: Cannot fork
```

- The following message appears on the console or in the logs:

```
...
EAL:   PCI memory mapped at 0x7ffae4a40000
PMD: eth_em_dev_init(): port_id 2 vendorID=0x8086 deviceID=0x100e
Using fpn_port 0x7ffae654c000 size=150576 (0M)
Killed
//usr/bin/fast-path.sh: error starting //usr/bin//fp-rte. Check logs for ↵
↵details.
```

At this point, the machine may have hung. Check the logs after reboot, especially if they contain something similar to:

```
...
fp-rte[5113]: Using fp_ebtables_vr_shared=0x7ffae63c2000 size=4352 (0M)
fp-rte[5113]: Using fp-tc-shared=0x7ffad976f000 size=524608 (0M)
kernel: [ 1022.485264] fp-rte invoked oom-killer: gfp_mask=0x2d2, order=0, ↵
↵oom_score_adj=0
kernel: [ 1022.485271] fp-rte cpuset=/ mems_allowed=0
```

---

**Note:** Look for error messages either on the console or in the logs. See *rsyslog* and *journalctl* sections for details regarding what can be found in the logs.

---

### Hints

- There is a problem with the available memory, the fast path process has been killed because available memory was getting too small. Typically, after hugepages allocation, the fast path tried to allocate memory and there was not enough free.
- Add more memory.
- Check the output from `/proc/meminfo`, especially the `MemFree` field. See *meminfo* section for details. **MemFree** estimates how much memory is free before starting the fast path.

## 1G hugepages problems

### Symptoms

- The following message appears on the console or in the logs:

```
sh: echo: I/O error
WARNING: Can not allocate 1 hugepages for fast path
         0 pages of size 1024 MB were allocated
```

### Hints

- It seems you enabled the support of 1G hugepages in the kernel boot command line (`hugepagesz=1G default_hugepagesz=1G`). The fast path starting script failed to allocate the required amount of hugepages.

## OVA startup fails

### Symptoms

- With VMware 6.0 and vSphere desktop client, starting Turbo Router VM from OVA file fails with the following message:

```
The OVF package is invalid and cannot be deployed.
```

See <https://kb.vmware.com/s/article/2151537>.

### Hints

- Use the vSphere HTML5 client (the desktop client is deprecated).
- Repackage the OVA file to use SHA1 hashing instead of the latest SHA256 using `ovftool` available at <https://www.vmware.com/support/developer/ovf/>.

```
# ovftool --shaAlgorithm=SHA1 /path/to/original/file.ova /path/to/new/file-
↳ sha1.ova
```

## SR-IOV problems

### Symptoms

- Starting a VM (with PCI passthrough in its conf) with libvirt fails, yielding:

```
error: unsupported configuration: host doesn't support passthrough of host_
↳ PCI devices
```

Your XML libvirt domain contains something like this:

```
<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0x0000' bus='0x83' slot='0x00' function='0x0' />
  </source>
</hostdev>
```

## Hints

- Your NIC and your motherboard must support SR-IOV, and the Linux kernel must have booted with appropriate options. Enable the Directed I/O parameter in the BIOS (Basic Input/Output System), and ensure “intel\_iommu=on” is provided in the kernel command line.

## Turbo Router hangs when starting with i40e devices

### Symptoms

- Starting Turbo Router with i40e devices in a VM hangs. Looking at the logs:

```
Jun 22 22:15:07 dut-vm fp-rte[14244]: /usr/bin/fp-rte --huge-dir=/dev/
↳hugepages -n 4 -l 4-39 --socket-mem 2292 -d librte_ext_crypto_multibuffer.
↳so -w 0000:00:04.0 -w 0000:00:05.0 -w 0000:00:06.0 -- -t c4=0/c5=0/c6=0/
↳c7=0/c8=0/c9=0/c10=0/c11=0/c12=0/c13=0/c14=0/c15=0/c16=0/c17=0/c18=0/c19=0/
↳c20=0/c21=0/c22=1/c23=1/c24=1/c25=1/c26=1/c27=1/c28=1/c29=1/c30=1/c31=1/
↳c32=1/c33=1/c34=1/c35=1/c36=1/c37=1/c38=1/c39=1 --nb-mbuf 262144 -- --max-
↳vr=16
Jun 22 22:15:07 dut-vm fp-rte[14244]: EAL: Detected 40 lcore(s)
Jun 22 22:15:07 dut-vm fp-rte[14244]: EAL: Detected 1 NUMA nodes
Jun 22 22:15:07 dut-vm fp-rte[14244]: EAL: Multi-process socket /var/run/dpdk/
↳rte/mp_socket
Jun 22 22:15:07 dut-vm fp-rte[14244]: EAL: Some devices want iova as va but
↳pa will be used because.. EAL: vfio-noiommu mode configured
Jun 22 22:15:07 dut-vm fp-rte[14244]: EAL: No free hugepages reported in
↳hugepages-1048576kB
Jun 22 22:15:07 dut-vm fp-rte[14244]: Based on DPDK 18.05.0-6WIND.0
Jun 22 22:15:07 dut-vm fp-rte[14244]: EAL: Probing VFIO support...
Jun 22 22:15:07 dut-vm fp-rte[14244]: EAL: VFIO support initialized
Jun 22 22:15:08 dut-vm fp-rte[14244]: EAL: WARNING: cpu flags constant
↳tsc=yes nonstop_tsc=no -> using unreliable clock cycles !
Jun 22 22:15:08 dut-vm fp-rte[14244]: EAL: PCI device 0000:00:04.0 on NUMA
↳socket -1
Jun 22 22:15:08 dut-vm fp-rte[14244]: EAL: Invalid NUMA socket, default to 0
Jun 22 22:15:08 dut-vm fp-rte[14244]: EAL: probe driver: 8086:1583 net_i40e
Jun 22 22:15:08 dut-vm fp-rte[14244]: EAL: using IOMMU type 8 (No-IOMMU)
```

## Hints

- The i40e hardware has a known issue with regards to INTX interrupts. A workaround has been implemented in the vfio-pci kernel driver to hide INTX support and force a fallback to MSIX. The workaround must be applied on both the VM side and the hypervisor side. Upstream patch: <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=450744051d20>

This issue can be checked by looking at the kernel logs. Without the patch, some interrupt ends up as an orphan:

```
[ 219.768519] i40e 0000:83:00.0: i40e_ptp_stop: removed PHC on ens260f0
[ 223.302710] vfio_ecap_init: 0000:83:00.0 hiding ecap 0x19@0x1d0
[ 224.810517] vfio_bar_restore: 0000:83:00.0 reset recovery - restoring bars
[ 227.330187] irq 47: nobody cared (try booting with the "irqpoll" option)
[ 227.330195] CPU: 22 PID: 0 Comm: swapper/22 Not tainted 4.4.0-127-generic
↪#153-Ubuntu
[ 227.330197] Hardware name: Intel Corporation S2600CWR/S2600CWR, BIOS.
↪SE5C610.86B.01.01.0019.101220160604 10/12/2016
[ 227.330199] 00000000000000086 754d6f2e166f11f1 ffff88086de03e60.
↪ffffffff814001c3
[ 227.330203] ffff880864613e00 ffff880864613ed4 ffff88086de03e88.
↪ffffffff810e0c33
[ 227.330205] ffff880864613e00 0000000000000000 000000000000002f.
↪ffff88086de03ec0
[ 227.330208] Call Trace:
[ 227.330210] <IRQ> [<ffffffff814001c3>] dump_stack+0x63/0x90
[ 227.330225] [<ffffffff810e0c33>] __report_bad_irq+0x33/0xc0
[ 227.330228] [<ffffffff810e0fc7>] note_interrupt+0x247/0x290
[ 227.330232] [<ffffffff810de0b2>] handle_irq_event_percpu+0x172/0x1e0
[ 227.330234] [<ffffffff810de15e>] handle_irq_event+0x3e/0x60
[ 227.330237] [<ffffffff810e154c>] handle_fasteoi_irq+0x9c/0x160
[ 227.330243] [<ffffffff810311f3>] handle_irq+0x23/0x30
[ 227.330249] [<ffffffff8185419b>] do_IRQ+0x4b/0xe0
[ 227.330252] [<ffffffff8185187f>] common_interrupt+0xbf/0xbf
[ 227.330253] <EOI> [<ffffffff816e06b7>] ? cpuidle_enter_state+0x157/0x2d0
[ 227.330261] [<ffffffff816e0867>] cpuidle_enter+0x17/0x20
[ 227.330265] [<ffffffff810c72b2>] call_cpuidle+0x32/0x60
[ 227.330267] [<ffffffff816e0849>] ? cpuidle_select+0x19/0x20
[ 227.330269] [<ffffffff810c7576>] cpu_startup_entry+0x296/0x360
[ 227.330275] [<ffffffff81052b02>] start_secondary+0x172/0x1b0
[ 227.330276] handlers:
[ 227.330282] [<ffffffffffc01d0230>] vfio_intx_handler [vfio_pci]
[ 227.330284] Disabling IRQ #47
```

But, with the patch, vfio-pci reports that it has hidden INTX support:

```
[ 215.389554] i40e 0000:83:00.0: i40e_ptp_stop: removed PHC on ens260f0
[ 224.501452] vfio-pci 0000:83:00.0: Masking broken INTx support
```

(continues on next page)

(continued from previous page)

```
[ 224.501522] vfio_ecap_init: 0000:83:00.0 hiding ecap 0x19@0x1d0
[ 226.191488] vfio_bar_restore: 0000:83:00.0 reset recovery - restoring bars
```

## 4.2.2 Networking Issues

### Ports synchronization problems

#### Symptoms

- No ports are displayed when calling `fp-cli iface`.

#### Hints

- If you are dealing with physical NIC: Check that your NIC is detected by Linux, using `lspci`. See *lspci* section for details.
- Check the output from `fast-path.sh config --display` and make sure your NIC is among the selected ethernet cards.

### No packets are forwarded

#### Symptoms

- No packets are forwarded.
- `fp-cli stats non-zero` shows no (or low) `IpForwDatagrams` stats.
- `fp-cli dpdk-port-stats <port>` shows no (or low) rx/tx packets stats.
- `ip -s link show <interface>` shows no (or low) rx/tx packets stats.
- `kill -USR1 $(pidof fp-rte)` (Intel and Arm only) shows no (or low) rx/tx packets stats.

#### Hints

- Check whether configurations between Linux and the fast path are consistent:
  - Check IP addresses and routes configured in the kernel, using `ip address show` and `ip route show`. Check whether the interfaces and bridges are up and running using `ip link show` and `brctl show <bridge_name>`.
- Check IP addresses and routes known to the fast path, using `fp-cli route4 type all`.
- If you are using bridges, check whether your bridges have correct states, using `fp-cli bridge`.
- Check that `fp_dropped` fast path statistics are not too high using `fp-cli stats percore non-zero`. A high `fp_dropped` stat suggests that packets are somehow not acceptable for the fast path. The ideal case is when forwarding stats are evenly spread throughout cores, that is when each core more or less forwards as many packets as the others. See *Fast Path statistics* section for an example of stats.

- Check that `exception stats` fast path statistics are not too high. Basic exceptions indicate how many packets could not be processed by the fast path, and have thus been injected in the linux stack for slow path processing. If the value is high, it is a good indicator that IP addresses/routes/tunnels in the fast path are badly configured. See *Fast Path statistics* section for an example of stats.
- Check whether it works correctly when the fast path is turned off. See *Turn Fast Path off* section for details.

## Netfilter synchronization problems

### Symptoms

- Packets are not filtered according to your iptables rules.

### Hints

- Check whether filtering rules between Linux and fast path are consistent:
  - Check filtering rules in the kernel, using `ip[6]tables -S`. Refer to the `ip[6]tables` manpage for details on this command.
  - Check filtering rules known to the fast path, using `fp-cli nf[4|6]-table <filter|mangle|nat> [all|nonzero]`. Check also whether the filtering module is enabled, using `fp-cli nf[4|6]`. Some targets and rules are not supported in the fast path: check that you are using only documented supported options.

## Connectivity problems

### Symptoms

- I can no longer connect (via the network) to my machine.
- The VM was configured to redirect connections to the guest (using something like `-netdev user, id=user.0,hostfwd=tcp::35047-:22`).

### Hints

- When starting Turbo Router, NIC kernel drivers have been unloaded and thus all IP configuration lost.

## Network configuration lost after restart

### Symptoms

- My network configuration no longer works after reboot or Turbo Router restart. For example:
  1. My linux bridge is empty after stopping (or restarting) the fast path:

```
# brctl show
bridge name      bridge id      STP enabled    interfaces
```

## Hints

- The fast path may replace, change, delete and create netdevices. Any tool (brctl, iproute2, etc.) that use “old” references to netdevices must have its configuration refreshed when the fast path is stopped.
  - For linux bridge, recreate the bridge and re-add the ports if need be. e.g.:

```
# brctl addbr br0
# brctl addif br0 eth1
# brctl addif br0 tap0
```

## DKMS takes too long

### Symptoms

- Modules recompilation/removal with DKMS takes too long.

### Hints

- Edit the DKMS configuration in `/etc/dkms/framework.conf`, to prevent it from running some long operations:

```
# mkdir -p /etc/dkms
# echo 'no_initrd="y"' >> /etc/dkms/framework.conf
# echo 'no_depmod="y"' >> /etc/dkms/framework.conf
```

- Disable weak-modules:

```
# chmod a-x /sbin/weak-modules
```

## VRRP is unable to work on VMware virtual machines

### Symptoms

- VRRP reports master state on all members but no member receives packets intended for the VRRP virtual IP

### Cause

**The VMware VSwitch drops frames to MAC addresses that are unknown from the network card properties of the**

- VRRP gives the ability to define a virtual IP that can move between machines. By design, the virtual IP is associated to a virtual MAC address - different from the real network card's MAC address. Using a virtual MAC address instead of a real MAC address makes the switchover quicker as no update of ARP tables is needed. However, a Virtual MAC address is not supported by the VMware VSwitch. Unlike physical switches, the VSwitch has no MAC learning mechanism capability. The network section of the VM properties defines virtual network cards and one MAC address per card.

The VSwitch only knows those addresses to determine on what port to send a frame. Frames to any unknown addresses, including virtual VRRP MAC addresses, are dropped.

- VRRP uses multicast packets to send VRRP protocol messages between its members. Multicast packets use typical multicast MAC addresses that are also not known by the VSwitch.

## Hints

### One of the following solutions should be applied:

- Warning: this solution may impact the performance on VMware hypervisor. Enable promiscuous mode on all VSwitches associated with the VLAN you want VRRP to run on. Basically, the VM will receive all traffic within the VSwitch VLAN. Refer to <https://kb.vmware.com/s/article/1002934> for more information.
- Set up the VRRP instance to disable the usage of a virtual MAC address “vmac” and to use manual unicast peers to exchange VRRP protocol data unit instead of using multicast. This solution is only applicable to VMware and should not be applied on any other context without an explicit request from support. In this mode, the virtual IP address is associated to the real NIC MAC address of the active member. Upon member switchover, a gratuitous ARP is sent to advertise other machines to update their ARP table with the new MAC. You must ensure and test that gratuitous ARP are treated correctly by all machines. If not, some machines would lose connectivity until the ARP cache timeout expires.

## Conflict between i40e FW-LLDP and software LLDP agents

### Symptoms

- LLDPDU are not received while the source correctly sends them and the link between both machine works.

### Cause

#### LLDPDU may be consumed by the LLDP engine integrated in the network card firmware:

- Some Intel network adapter (like Ethernet 700 series) has built-in hardware LLDP engine, which is enabled by default. The LLDP Engine is responsible for receiving and consuming LLDP frames, and also replies to the LLDP frames that it receives. The LLDP engine does not forward LLDP frames to the network stack of the Operating System. The i40e driver enable this feature by default.

## Hints

### The firmware LLDP must be disabled in i40e ports:

- For fast path ports:

```
# fp-cli dpdk-i40e-debug-lldp-cmd on|off
```

- For Linux ports: To disable the FW-LLDP:

```
# ethtool --set-priv-flags <ifname> disable-fw-lldp off
```

To check the FW-LLDP setting:

```
# ethtool --show-priv-flags <ethX>
```

**See also:**

The FW-LLDP (Firmware Link Layer Discovery Protocol) chapter of the [i40e Linux Base Driver for Intel controller](https://downloadmirror.intel.com/24411/eng/README.txt) (<https://downloadmirror.intel.com/24411/eng/README.txt>).

## 4.2.3 Performance Tuning

### Slow packet processing

#### Symptoms

- Packet processing performance is not as high as expected.

#### Hints

- Follow the advice provided in `fast-path.sh config -i` when using the advanced configuration.
- If running in a VM, check that the `qemu` instance handling your VM is pinned on specific cores. See *CPU Pinning for VMs* section for details.

### Performance drop with Mellanox ConnectX-3 devices

#### Symptoms:

- Packet processing is slower than expected

#### Hints:

- On Dell and SuperMicro servers, PCI read buffer may be misconfigured for ConnectX-3/ConnectX-3-Pro NICs. Check the output of `setpci -s <NIC_PCI_address> 68.w`. For instance:

```
# lspci | grep Mellanox
04:00.0 Ethernet controller: Mellanox Technologies MT27520 Family [ConnectX-3-
↔Pro]
# setpci -s 04:00.0 68.w
202e
```

**Warning:** Beware with the following command, it is known to cause spontaneous reboot on some systems.

If the value is below 0x5020 (here that’s the case), set it to 0x5020:

```
# setpci -s 04:00.0 68.w=5020
```

### Performance drop with AWS ENA devices

#### Symptoms:

- Packet processing is slower than expected

#### Hints:

- If ENAv2 is used (both HW, and SW driver), the Low Latency Queue v2 (LLQv2) feature may be enabled on the platform. In this case, [DPDK documentation](https://doc.dpdk.org/guides/nics/ena.html#prerequisites) suggests (<https://doc.dpdk.org/guides/nics/ena.html#prerequisites>) enabling “write combining” for the virtualization driver used as a backend by the PMD. This is disabled by default.

You can automatically do this at module insertion time with an appropriate modprobe.d configuration file:

```
# /etc/modprobe.d/igb_uio.conf
options igb_uio wc_activate=1
```

**Warning:** Be careful though, as this has been known to degrade performance depending on your setup (kernel version, ENA without LLQv2...), so make sure to test it.

## 4.2.4 OpenStack

This section gathers issues that happen with Turbo Router started in an OpenStack environment.

### VM start errors

#### Symptoms

- My VM can’t start, or is in a bad state (NOSTATE):

```
$ nova list
```

| ID                                   | Name | Status | Task State | Power   |
|--------------------------------------|------|--------|------------|---------|
| 52ad953d-19dd-47a9-b03d-dfe565e655e1 | vm3  | ERROR  | -          | NOSTATE |

(continues on next page)

(continued from previous page)

```

| b28e5aa1-05c9-494b-8f0e-0247d95bde87 | vm2 | ACTIVE | - | Running
↪ | private2=12.0.0.3 |
| c4a52ed6-775d-45b3-96c2-8c2a6a1530ac | vm1 | ACTIVE | - | Running
↪ | private=11.0.0.6, 172.24.4.3 |
+-----+-----+-----+-----+
↪-----+-----+-----+-----+

```

**Hints**

- Check the /var/log/nova/nova-compute.log file for ERROR. Considering the output, check the following issues.

**Not enough memory**

**Symptoms** My VM can't start, or is in a bad state (NOSTATE). On the compute node hosting the VM. /var/log/nova/nova-compute.log shows ERRORS and TRACES like those:

```

Error launching a defined domain with XML: <domain type='kvm'>
[instance: 52ad953d-19dd-47a9-b03d-dfe565e655e1] Instance failed to spawn
...
...: unable to map backing store for hugepages: Cannot allocate memory

```

**Hints**

- Add more memory to your compute node.

**Cannot use hugepages of 1GB**

**Symptoms** Nova displays an error “Unable to find any usable hugetlbfs mount”.

On the controller node, /var/log/nova/nova-conductor.log shows ERRORS and TRACES like this one:

```

error: Unable to find any usable hugetlbfs mount for 1048576 KiB

```

**Hints**

- Hugepages cannot be allocated for the VM. It may be due to the size of the hugepages. Try to allocate more but smaller hugepages.

## Performance degradation and security groups

### Symptoms

- VM packet processing is slower than expected.

### Hints

- Consider disabling security groups as numerous packets processing require many iptables/ebtables look-ups to direct packets properly when they're enabled.

Refer to OpenStack documentation on how to do that considering your running version.

## 4.2.5 Management

### SNMP process is crashing in case of full-route

#### Symptoms

- SNMP process crashes during SNMP walking when the router has loaded the full internet routing table.

#### Hints

- Blacklist route table OID so that SNMP process is not overloaded.
- Add the following SNMP configuration:

```
/ vrf main snmp view <view> subtree .1 included true
/ vrf main snmp view <view> subtree .1.3.6.1.2.1.4.21 included false
/ vrf main snmp view <view> subtree .1.3.6.1.2.1.4.24 included false
```

## 4.3 Fast Path Information

### 4.3.1 Fast Path statistics

Use `fp-cli stats [percore] [non-zero]` to get the statistics recorded by the fast path:

```
# fp-cli stats non-zero
==== interface stats:
lo-vr0 port:254
mgmt0-vr0 port:254
enp3s0f1-vr0 port:254
ens785f1-vr0 port:254
ens787f1-vr0 port:254
ens804f1-vr0 port:254
ens806f1-vr0 port:254
```

(continues on next page)

(continued from previous page)

```

fnp0-vr0 port:254
ntfp4-vr0 port:0
ntfp1-vr0 port:1
ntfp2-vr0 port:2
ntfp3-vr0 port:3
==== global stats:
==== exception stats:
  LocalBasicExceptions:7
  LocalExceptionClass:
  LocalExceptionType:
==== IPv4 stats:
  IpForwDatagrams:509870627
  IpInReceives:509870627
==== arp stats:
==== IPv6 stats:
==== TCP stats:
==== UDP stats:
==== UDP6 stats:
==== IPsec stats:
==== IPsec IPv6 stats:
==== L2 stats:
==== fp-vswitch stats:

```

### 4.3.2 fp-cpu-usage

Use this command to get the fast path usage per core, and the number of cycles to process one packet:

```

# fp-cpu-usage
Fast path CPU usage:
cpu: %busy    cycles    cycles/packet
  2:  70%  227166479    990
  3:  69%  222733174    991
...
average cycles/packets received from NIC: 991 (5389132282/5436242)

```

It is a good indicator regarding how busy the fast path cores are, processing packets.

### 4.3.3 Turn Fast Path off

Use the following command to turn most of the fast path off:

```
# fp-cli fp-state-set off
FP is stopped (was started)
```

By doing this, no processing will be done by the fast path. As soon as the fast path receives a packet on a port, without any processing, it will inject it in the linux stack.

If the test works with the fast path thus disabled, it usually means the fast path drops packets.

## 4.4 System Information

### 4.4.1 CPU Pinning for VMs

For each virtual CPU, QEMU uses one pthread for actually running the VM and pthreads for management. For best performance, you need to make sure cores used to run fast path dataplane are only used for that.

To get the threads associated with each virtual CPU, use `info cpus` in QEMU monitor console:

```
QEMU 2.3.0 monitor - type 'help' for more information
(qemu) info cpus
* CPU #0: pc=0xffffffff8104f596 (halted) thread_id=26773
  CPU #1: pc=0x000007faee19be9f9 thread_id=26774
  CPU #2: pc=0xffffffff8104f596 (halted) thread_id=26775
  CPU #3: pc=0x000000000000530233 thread_id=26776
```

To get all threads associated with your running VM (including management threads) and what CPU they are currently pinned on, call:

```
# taskset -ap <qemu_pid>
pid 26770's current affinity mask: f
pid 26771's current affinity mask: f
pid 26773's current affinity mask: f
pid 26774's current affinity mask: f
pid 26775's current affinity mask: f
pid 26776's current affinity mask: f
pid 27053's current affinity mask: f
```

By pinning our VM on a specific set of cores, we ensure less overload.

You may either run `qemu` with a specific set of cores when starting, using:

```
# taskset -c 0-1 <qemu command>
```

You may also pin a VM after it has been started, using the PID (Process Identifier) of its threads. For instance, to change the physical CPU on which to pin the virtual CPU #0, use:

```
# taskset -cp 0-1 26773
pid 26773's current affinity list: 0-3
pid 26773's new affinity list: 0,1
```

---

**Note:** Refer to the `taskset` manpage for specific options.

---

When using `libvirt`, you may use `<cputune>` with `vcupin` to pin virtual CPUs to physical ones. e.g.:

```
<vcpu cpuset='7-8,27-28'>4</vcpu>
<cputune>
  <vcupin vcpu="0" cpuset="7"/>
  <vcupin vcpu="1" cpuset="8"/>
  <vcupin vcpu="2" cpuset="27"/>
  <vcupin vcpu="3" cpuset="28"/>
</cputune>
```

---

**Note:** Refer to the `libvirt Domain XML format` (<http://libvirt.org/formatdomain.html#elementsCPUTuning>) documentation for further details.

---

We can look at `htop` results (after filtering results for this `qemu` instance) to confirm what threads are actually used:

| PID   | USER  | VRT   | RES   | SHR  | S | CPU% | MEM% | TIME+   | NLWP | Command  |
|-------|-------|-------|-------|------|---|------|------|---------|------|--|
| 26770 | mazon | 7032M | 4067M | 7092 | S | 200. | 25.5 | 2h19:55 | 7    | - qemu-system-x86_64 -<br>↳daemonize --enable-kvm -m 6G -cpu host -smp sockets=1,cores=4,threads=1 ... |
| 27053 | mazon | 7032M | 4067M | 7092 | S | 0.0  | 25.5 | 0:01.13 | 7    | - qemu-system-x86_64 -<br>↳daemonize --enable-kvm -m 6G -cpu host -smp sockets=1,cores=4,threads=1 ... |
| 26776 | mazon | 7032M | 4067M | 7092 | R | 99.1 | 25.5 | 1h04:38 | 7    | - qemu-system-x86_64 -<br>↳daemonize --enable-kvm -m 6G -cpu host -smp sockets=1,cores=4,threads=1 ... |
| 26775 | mazon | 7032M | 4067M | 7092 | S | 0.9  | 25.5 | 2:48.21 | 7    | - qemu-system-x86_64 -<br>↳daemonize --enable-kvm -m 6G -cpu host -smp sockets=1,cores=4,threads=1 ... |
| 26774 | mazon | 7032M | 4067M | 7092 | R | 99.1 | 25.5 | 1h09:42 | 7    | - qemu-system-x86_64 -<br>↳daemonize --enable-kvm -m 6G -cpu host -smp sockets=1,cores=4,threads=1 ... |
| 26773 | mazon | 7032M | 4067M | 7092 | S | 0.0  | 25.5 | 2:23.03 | 7    | - qemu-system-x86_64 -<br>↳daemonize --enable-kvm -m 6G -cpu host -smp sockets=1,cores=4,threads=1 ... |
| 26771 | mazon | 7032M | 4067M | 7092 | S | 0.0  | 25.5 | 0:00.00 | 7    | - qemu-system-x86_64 -<br>↳daemonize --enable-kvm -m 6G -cpu host -smp sockets=1,cores=4,threads=1 ... |

You may even change CPU affinity by typing a `w` when on a specific PID line in `htop`.

Similarly, you can get threads PID by looking in `/proc/<pid>/task/`, e.g.:

```
# ls /proc/26773/task
26770/ 26771/ 26773/ 26774/ 26775/ 26776/ 27053/
```

#### 4.4.2 fp-cli dpdk-port-stats

The `fp-cli dpdk-port-stats` command is used to display and set options related to network drivers (for those that support it).

To display the statistics for a given port, use `fp-cli dpdk-port-stats <port>`:

```
# fp-cli dpdk-port-stats <port>

rx_good_packets: 261064663
tx_good_packets: 256512062
rx_good_bytes: 15663879780
tx_good_bytes: 15390725600
rx_missed_errors: 0
rx_errors: 36554346
tx_errors: 0
rx_mbuf_allocation_errors: 0
rx_q0_packets: 32632951
rx_q0_bytes: 1957977060
rx_q0_errors: 0
...
tx_q0_packets: 128251039
tx_q0_bytes: 7695062334
...
rx_total_packets: 297619011
rx_total_bytes: 17857140760
tx_total_packets: 256512062
tx_size_64_packets: 256512046
...
```

The most important stats to look at are the `{r,t}x_good_{packets,bytes}` and errors.

They indicate globally how well the port is handling packets.

There is also per queue statistics that might be interesting in case of multiqueue. It's better to have packets transmitted on as many different queues as possible, but it depends on various factors, such as the IP addresses and UDP / TCP ports.

The drop statistics provide useful information about why packets are dropped. For instance, the `rx_missed_errors` counter represents the number of packets dropped because the CPU was not fast enough to dequeue them. A non-zero value for `rx_mbuf_allocation_errors` shows that there is not enough mbuf structure configured in the fast path.

---

**Note:** Statistics field names may vary considering the driver.

---

`fp-cli dpdk-port-offload` can be used to check whether offload is enabled, using the following:

```
# fp-cli dpdk-port-offload <port>

TX vlan insert off [fixed]
TX IPv4 checksum off [fixed]
TX TCP checksum off [fixed]
TX UDP checksum off [fixed]
TX SCTP checksum off [fixed]
TX TSO off [fixed]
TX UFO off [fixed]
RX vlan strip off
RX vlan filter off
RX IPv4 checksum on
RX TCP checksum on
RX UDP checksum on
RX MPLS IP off
RX LRO off
RX GRO off
```

If you want to enable TSO (TCP Segmentation Offload) (which should provide you with better performance for TCP, as the hardware will handle the segmentation), use:

```
# fp-cli dpdk-port-offload-set eth1 tso on
```

---

**Note:** You can get various error messages when trying to change hardware parameters. For instance, `Cannot change tcp-segmentation-offload` may appear if the driver does not support to dynamically change TSO offload.

---

### 4.4.3 `lspci`

The `lspci` command is useful to display information about PCI buses. In most cases, we look for “Ethernet” devices.

Use `lspci` to get basic information on all connected devices:

```
# lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]
```

(continues on next page)

(continued from previous page)

```
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIIX4 ACPI (rev 03)
00:02.0 VGA compatible controller: Device 1234:1111 (rev 02)
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller
↳(rev 03)
```

To display the driver handling devices, use:

```
# lspci -k
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
    Subsystem: Red Hat, Inc Qemu virtual machine
00:01.0 ISA bridge: Intel Corporation 82371SB PIIIX3 ISA [Natoma/Triton II]
    Subsystem: Red Hat, Inc Qemu virtual machine
00:01.1 IDE interface: Intel Corporation 82371SB PIIIX3 IDE [Natoma/Triton II]
    Subsystem: Red Hat, Inc Qemu virtual machine
    Kernel driver in use: ata_piix
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIIX4 ACPI (rev 03)
    Subsystem: Red Hat, Inc Qemu virtual machine
00:02.0 VGA compatible controller: Device 1234:1111 (rev 02)
    Subsystem: Red Hat, Inc Device 1100
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller
↳(rev 03)
    Subsystem: Red Hat, Inc QEMU Virtual Machine
    Kernel driver in use: igb_uio
```

---

**Note:** Refer to the `lspci` manpage for specific options.

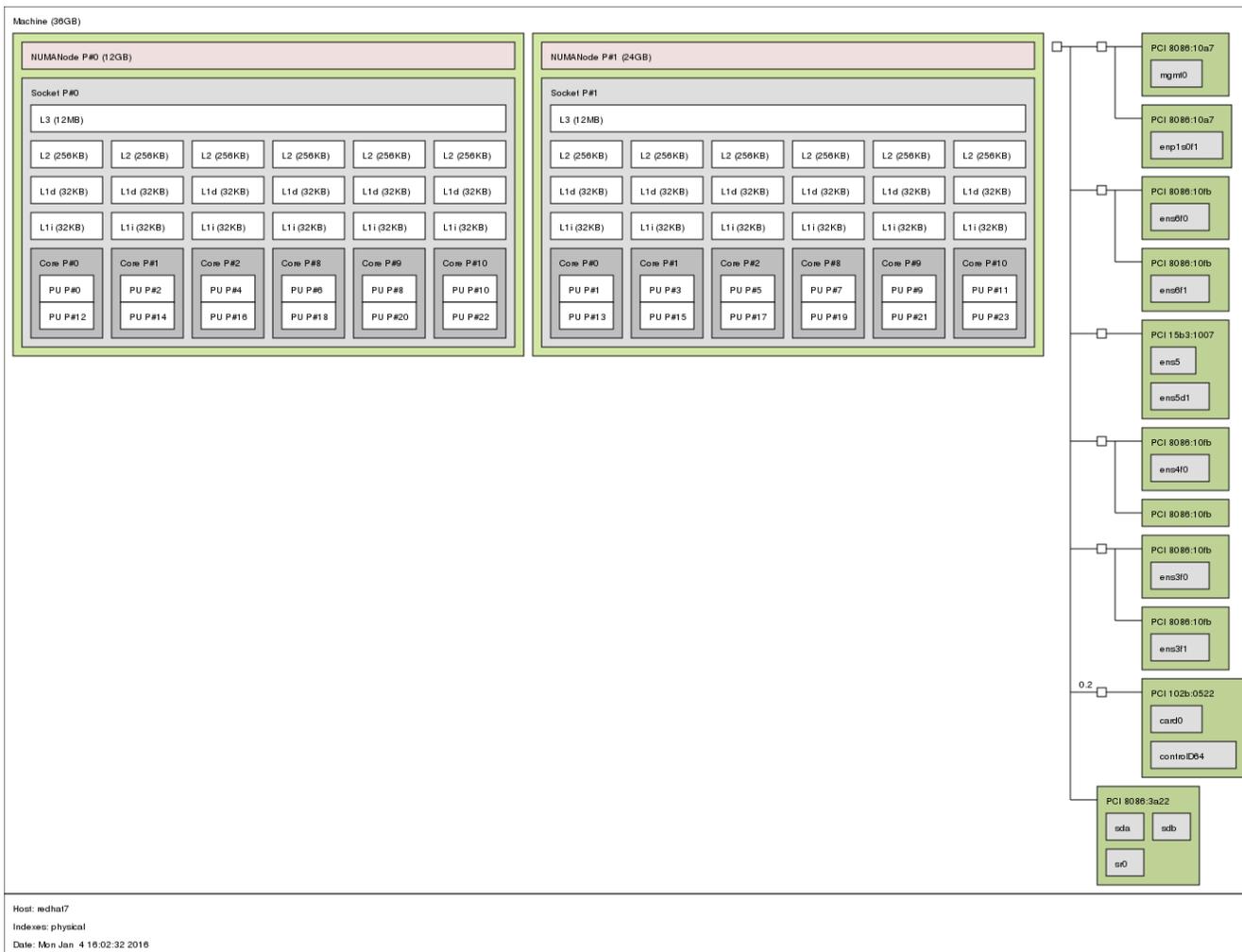
---

#### 4.4.4 `lstopo`

`lstopo` provides a global view of the system's topology. It details what machines contain what nodes, containing sockets, containing cores, containing processor units.

The following command presents a graphical representation of a big machine's topology:

```
# lstopo --of png > lstopo_output.png
```



You can use the following command to get a textual representation:

```
# lstopo --of txt
```

### 4.4.5 meminfo

The file `/proc/meminfo` presents a memory status summary. You can also look at memory by node through `/sys/devices/system/node/node<node_id>/meminfo`.

On a VM with 1GB of RAM (Random-Access Memory) running redhat-7, we have this:

```
# cat /proc/meminfo
MemTotal:      1016548 kB
MemFree:       107716 kB
MemAvailable:  735736 kB
Buffers:       83244 kB
```

(continues on next page)

(continued from previous page)

|                    |                |
|--------------------|----------------|
| Cached:            | 626528 kB      |
| SwapCached:        | 0 kB           |
| Active:            | 400416 kB      |
| Inactive:          | 352892 kB      |
| Active(anon):      | 49808 kB       |
| Inactive(anon):    | 13304 kB       |
| Active(file):      | 350608 kB      |
| Inactive(file):    | 339588 kB      |
| Unevictable:       | 0 kB           |
| Mlocked:           | 0 kB           |
| SwapTotal:         | 0 kB           |
| SwapFree:          | 0 kB           |
| Dirty:             | 0 kB           |
| Writeback:         | 0 kB           |
| AnonPages:         | 43652 kB       |
| Mapped:            | 9500 kB        |
| Shmem:             | 19572 kB       |
| Slab:              | 130972 kB      |
| SReclaimable:      | 100896 kB      |
| SUnreclaim:        | 30076 kB       |
| KernelStack:       | 1888 kB        |
| PageTables:        | 2692 kB        |
| NFS_Unstable:      | 0 kB           |
| Bounce:            | 0 kB           |
| WritebackTmp:      | 0 kB           |
| CommitLimit:       | 507248 kB      |
| Committed_AS:      | 214004 kB      |
| VmallocTotal:      | 34359738367 kB |
| VmallocUsed:       | 4412 kB        |
| VmallocChunk:      | 34359730912 kB |
| HardwareCorrupted: | 0 kB           |
| AnonHugePages:     | 4096 kB        |
| HugePages_Total:   | 1              |
| HugePages_Free:    | 0              |
| HugePages_Rsvd:    | 0              |
| HugePages_Surp:    | 0              |
| Hugepagesize:      | 2048 kB        |
| DirectMap4k:       | 79744 kB       |
| DirectMap2M:       | 968704 kB      |

**Note:** The kernel documentation provides details regarding meminfo [here](http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Documentation/filesystems/proc.txt?id=HEAD#n819) (<http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Documentation/filesystems/proc.txt?id=HEAD#n819>).

For details regarding the HugePages fields, look [there](http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Document) (<http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Document>)

The most interesting fields in our case are:

**MemTotal** should be the same as the “total” memory displayed on top lines when running `top`

**MemFree** should be the same as the “free” memory displayed on top lines when running `top`

**MemAvailable** estimate of how much memory is available for starting new applications, without swapping

**HugePages\_Total** size of the pool of huge pages

**HugePages\_Free** number of huge pages in the pool that are not yet allocated

**HugePages\_Rsvd** number of huge pages for which a commitment to allocate from the pool has been made, but no allocation has yet been made

### 4.4.6 numastat

This tool shows per-NUMA-node memory statistics for processes and the operating system.

Without argument, it displays per-node NUMA hit and miss system statistics from the kernel memory allocator. A high value in `other_node` means that there are cross-NUMA memory transfers, which impacts performance. This information is dynamic and can be monitored with the `watch` command.

```
# numastat
                node0          node1
numa_hit          589246433      556912817
numa_miss           0             0
numa_foreign       0             0
interleave_hit     11616         17088
local_node         589229023      556900289
other_node         17410         12528
```

When a PID or a pattern is passed, it shows per-node memory allocation information for the specified process (including all its pthreads). The hugepages correspond to the DPDK memory, and the private area mainly corresponds to the shared memories.

```
# numastat fp-rte
Per-node process memory usage (in MBs) for PID 2176 (fp-rte:8)
                Node 0          Node 1          Total
-----
Huge             842.00         842.00         1684.00
Heap              0.41             0.00             0.41
Stack             0.03             0.00             0.03
Private          1004.35           24.27           1028.62
-----
Total            1846.79           866.27           2713.06
```

---

**Note:** Refer to the numa manpage for details.

---

## 4.5 Log Management

### 4.5.1 rsyslog

The rsyslogd daemon writes syslog messages in various places (considering its configuration in `/etc/rsyslog.conf`). Messages for the “daemon” facility are usually stored in `/var/log/daemon.log` (this is the case for buildroot images). However all messages are usually stored in `/var/log/syslog` (all facilities included), too.

The fast path sends syslog messages that you can look at later to figure out what happened during startup (and runtime). Here is an extract from `/var/log/syslog`:

```
fp-rte[3660]: EAL: Master lcore 1 is ready (tid=c4fdc300;cpuset=[1])
fp-rte[3660]: EAL: PCI device 0000:00:04.0 on NUMA socket -1
fp-rte[3660]: EAL: probe driver: 8086:100e rte_em_pmd
fp-rte[3660]: EAL: PCI memory mapped at 0x7f22c3c00000
fp-rte[3660]: PMD: eth_em_dev_init(): port_id 0 vendorID=0x8086 deviceID=0x100e
...
fp-rte[3660]: libfpm_shmem: write procfs: File exists
fp-rte[3660]: FPN: fp_mask=0x2 l_mask=0x2 dpvi_mask=0x1 stats_mask=0xd online=0xf
...
fp-rte[3660]: Create a mbuf pool on socket 0, nb_mbufs=16384
fp-rte[3660]: Bus Device ID Port# RXQ RXD/Q TXQ TXD/Q Excl
↳Interface Driver name
fp-rte[3660]: PCI 0000:00:04.0 8086:100e 0 1 128 1 512 1 N/A
↳ rte_em_pmd
fp-rte[3660]: PCI 0000:00:05.0 8086:100e 1 1 128 1 512 1 N/A
↳ rte_em_pmd
fp-rte[3660]: PCI 0000:00:06.0 8086:100e 2 1 128 1 512 1 N/A
↳ rte_em_pmd
fp-rte[3660]: Initializing port 0... ntxq=1 nrxq=1 [de:ed:01:f0:95:88] txq0=c1 rxq0=c1
↳PMD: eth_em_tx_queue_setup(): sw_ring=0x7f22acdccc00 hw_ring=0x7f22acdced00 dma_
↳addr=0xaffced00
fp-rte[3660]: PMD: eth_em_rx_queue_setup(): sw_ring=0x7f22acdbc6c0 hw_
↳ring=0x7f22acdbcbc0 dma_addr=0xaffbcbc0
fp-rte[3660]: PMD: eth_em_rx_init(): forcing scatter mode
fp-rte[3660]: PMD: eth_em_start(): <<
fp-rte[3660]: done
fp-rte[3660]: Initializing port 1... ntxq=1 nrxq=1 [de:ed:02:f7:f2:e5] txq0=c1 rxq0=c1
↳PMD: eth_em_tx_queue_setup(): sw_ring=0x7f22acdaa480 hw_ring=0x7f22acdac580 dma_
↳addr=0xaffac580
```

(continues on next page)

(continued from previous page)

```

fp-rte[3660]: PMD: eth_em_rx_queue_setup(): sw_ring=0x7f22acd99f40 hw_
↳ring=0x7f22acd9a440 dma_addr=0xaff9a440
fp-rte[3660]: PMD: eth_em_rx_init(): forcing scatter mode
fp-rte[3660]: PMD: eth_em_start(): <<
fp-rte[3660]: done
...
fp-rte[3660]: fpn_sdk_init: dedicated configuration polling lcore -1
fp-rte[3660]: fpn_dpvi_shmem_mmap: fpn_dpvi_shmem sizeof=80
fp-rte[3660]: fpn_dpvi_ring_shmem_mmap: fpn_dpvi_ring_shmem size=266496
fp-rte[3660]: fpn_per_lcore_dpvi_shmem_init: lcoreid 1 mbufs=768
kernel: [ 57.450256] dpvi: kernel_cpumask_display() dpvi: fp_mask = 0x2
kernel: [ 57.450259] dpvi: kernel_cpumask_display() dpvi: dpvi_mask = 0x1
kernel: [ 57.450260] dpvi: kernel_cpumask_display() dpvi: l_mask = 0x2
kernel: [ 57.450261] dpvi: kernel_cpumask_display() dpvi: online_mask = 0xf
kernel: [ 57.450263] dpvi: dpvi_init_ring() dpvi: cpu 0 use Tx queue 0 ring 1
kernel: [ 57.450264] dpvi: dpvi_init_ring() dpvi: cpu 1 use Tx queue 0 ring 1
kernel: [ 57.450264] dpvi: dpvi_init_ring() dpvi: cpu 2 use Tx queue 0 ring 1
kernel: [ 57.450265] dpvi: dpvi_init_ring() dpvi: cpu 3 use Tx queue 0 ring 1
kernel: [ 57.451284] dpvi: dpvi_sysctl_running_fastpath() Watching PID 3660
fp-rte[3660]: fpn-sdk init finished
fp-rte[3660]: fp-ovs: using accelerated functions(avx[x] sse4.2[x] sse4.1[x])
fp-rte[3660]: Using fp-shared=0x7f22a0069000 size=20390912 (19M)
...
fp-rte[3660]: fp-ovs: using accelerated functions(avx[x] sse4.2[x] sse4.1[x])
fp-rte[3660]: fp-vswitch module loaded
fp-rte[3660]: Init core 1
fp-rte[3660]: entering main loop on lcore 1 (master)
fp-rte[3660]: RX -- lcoreid=1 queueid=0 portid=0
fp-rte[3660]: RX -- lcoreid=1 queueid=0 portid=1
fp-rte[3660]: RX -- lcoreid=1 queueid=0 portid=2
fp-rte[3660]: TX -- lcoreid=1 queueid=0 portid=0
fp-rte[3660]: TX -- lcoreid=1 queueid=0 portid=1
fp-rte[3660]: TX -- lcoreid=1 queueid=0 portid=2

```

If you don't see anything in `/var/log/syslog`, make sure the `rsyslogd` daemon is running:

```

# ps aux | grep syslog
root      83  0.0  0.0 251864 2648 ?        Ssl  13:23   0:00 /usr/sbin/rsyslogd
root     851  0.0  0.0  4676   648 ttyS0    R+   14:32   0:00 grep syslog

```

If `rsyslogd` is not running, refer to your distribution documentation on how to get it started.

---

**Note:** Refer to the appropriate manpage (e.g.: `man rsyslog.conf`) for configuration options.

---

## 4.5.2 journalctl

With SystemD, logging is handled by the `systemd-journald` daemon. It writes its log in a binary format, and one usually uses `journalctl` to access it.

Use this command to see syslog messages from a given program (providing its path):

```
# journalctl /usr/bin/fpmd
Dec 10 15:56:44 dut-vm fpmd[11412]: bpf module registered
Dec 10 15:56:44 dut-vm fpmd[11412]: inaddr module registered
Dec 10 15:56:44 dut-vm fpmd[11412]: inroute module registered
...
```

You can combine it to follow logs from several programs at once. e.g.:

```
# journalctl /usr/bin/cmgrd /usr/bin/fpmd
...
Dec 10 15:56:44 dut-vm fpmd[11412]: tunnel module registered
Dec 10 15:56:44 dut-vm fpmd[11412]: fpm_netlink_recv: fpn0 found : ifindex 6 status 40
Dec 10 15:56:44 dut-vm cmgrd[11678]: fp-vswitch module loaded
Dec 10 15:56:44 dut-vm cmgrd[11678]: fpm_connect: trying to connect to fpm
Dec 10 15:56:44 dut-vm cmgrd[11678]: fpvs_cm_init_cb: Could not get OVS "ovs_datapath"
↳family info
Dec 10 15:56:44 dut-vm fpmd[11413]: add:cannot set flags in FP ens4-vr0: [-95]
...
```

---

**Note:** Refer to the appropriate manpage (e.g.: `man journalctl`) for configuration options.

---

## 4.5.3 fast path logs

If you have an issue when starting the fast path, take a look at the `/var/log/fast-path.log` file for fast path startup log messages.

For instance (on a normal startup):

```
# cat /var/log/fast-path.log
Starting Fast Path...
/usr/bin/fp-rte --huge-dir=/mnt/huge -n 4 -l 5-6,25-26 --socket-mem 438,0 -d librte_
↳pmd_vhost.so -w 0000:05:00.0 -w 0000:05:00.1 -w 0000:05:00.2 -w 0000:05:00.3 -w
↳0000:83:00.0 -w 0000:83:00.1 --vdev
=pmd-vhost0,sockname=/tmp/pmd-vhost0,rxqmap=auto:rr/nb_ring:1,txqmap=auto:hash/nb_
↳ring:1,loglevel=2 --vdev=pmd-vhost1,sockname=/tmp/pmd-vhost1,rxqmap=auto:rr/nb_
↳ring:1,txqmap=auto:hash/nb_ring:1,loglevel=2
-- -t c5=0:1:2:3:4:5/c6=0:1:2:3:4:5/c25=0:1:2:3:4:5/c26=0:1:2:3:4:5 --nb-mbuf 65536,0
↳
```

(continues on next page)

(continued from previous page)

```

Based on DPDK v2.2.0
EAL: Detected lcore 0 as core 0 on socket 0
...
EAL: Detected lcore 39 as core 12 on socket 1
EAL: Support maximum 255 logical core(s) by configuration.
EAL: Detected 40 lcore(s)
EAL: VFIO modules not all loaded, skip VFIO support...
EAL: Setting up physically contiguous memory...
...
EAL: Requesting 219 pages of size 2MB from socket 0
EAL: TSC frequency is ~2992788 KHz
EAL: open shared lib librte_pmd_vhost.so
PMD: PMD virtio vhost, Copyright(c) 2014-2015 6WIND S.A.
EAL: Master lcore 5 is ready (tid=aa486b40;cpuset=[5])
PMD: Initializing 6WIND vhost PMD (pmd-vhost0)
PMD: pmd-vhost[0] pmd_vhost_parse_args_cb(): The loglevel option is deprecated. Please,
↳ use the verbose option to enable debug messages
PMD: Initializing 6WIND vhost PMD (pmd-vhost1)
PMD: pmd-vhost[0] pmd_vhost_parse_args_cb(): The loglevel option is deprecated. Please,
↳ use the verbose option to enable debug messages
EAL: lcore 25 is ready (tid=8bbe7700;cpuset=[25])
EAL: lcore 6 is ready (tid=8c3e8700;cpuset=[6])
EAL: lcore 26 is ready (tid=8b3e6700;cpuset=[26])
EAL: PCI device 0000:05:00.0 on NUMA socket 0
EAL: probe driver: 8086:1521 rte_igb_pmd
EAL: PCI memory mapped at 0x7fb1a9000000
EAL: PCI memory mapped at 0x7fb1a9020000
PMD: eth_igb_dev_init(): port_id 2 vendorID=0x8086 deviceID=0x1521
...
Using fpn_port 0x7fb1aa327000 size=150576 (0M)
Starting cpuset...
cpuset.sh: creating cpuset system
cpuset.sh: try to move all tasks from cpuset root to system
.....
↳.....cpuset.sh: moved 596 tasks among 1204
cpuset successfully started
Info: no configuration file /etc/fp-daemons.env for fp-daemons.sh, using defaults
Starting Fast Path Daemons...
Starting Fast Path Manager...
/usr/bin/fpmd
Fast Path Manager successfully started
Starting Hitflags daemon...
/usr/bin/hitflagsd
Hitflags daemon successfully started

```

(continues on next page)

(continued from previous page)

```
Fast Path Daemons successfully started
Fast Path successfully started
```

#### 4.5.4 fpmd logs

You can start `fpmd` with `-v` option to have more verbose output. To do that, either:

1. kill and restart (providing `-v`) the `fpmd` process, once the fast path has been started:

```
# killall fpmd
# fpmd -v
```

2. provide `-v` in `FPM_OPTIONS` when starting the fast path (you could set it in `/etc/fpm.env` too):

```
# FPM_OPTIONS='-v' fast-path.sh start
```

#### See also:

*Linux - Fast Path Synchronization*

#### 4.5.5 cmgrd logs

When facing a synchronization issue, check first the line with `fpm_connection` in the log. It should display `connected to fpm socket` when the connection is established between `cmgrd` and `fpmd`.

You can start `cmgrd` with `-d` option (use `0xffffffff` for maximum debug level) to display more information regarding received netlink messages, messages sent to `fpmd`, etc. To do that, either:

1. kill and restart (providing `-d`) the `cmgrd` process, once the *Linux - Fast Path Synchronization* has been started:

```
# killall cmgrd
# cmgrd -d 0xffffffff
```

2. provide `-d` in `CMGR_OPTIONS` when starting the *Linux - Fast Path Synchronization* (you could set it in `/etc/cmgr.env` too):

```
# CMGR_OPTIONS='-d 0xffffffff' linux-fp-sync.sh start
```

#### See also:

*Linux - Fast Path Synchronization*

## 4.5.6 OpenStack logs

When you have an issue regarding VM spawning, take a look at `/var/log/nova/nova-compute.log` on the compute node hosting the VM.

In particular, look for messages with `error` or `trace` in it. For instance:

```
# grep -iE "(error|trace)" /var/log/nova/nova-compute.log
2016-01-27 11:37:22.286 12945 ERROR nova.network.linux_net [req-b7fdc659-2fd5-4d9e-
↪942c-803f71c2cce1 d82509fae77e41009880defd0bbd829e d9c0a5bd157947bab06d355bf4772db7 -
↪ - -] \
  Unable to execute ['ovs-vsctl', '--timeout=120', '--', '--if-exists', 'del-port', u
↪'tap13d2cb29-d6', '--', 'add-port', 'br-int', u'tap13d2cb29-d6', '--', 'set',
↪'Interface', \
      u'tap13d2cb29-d6', u'external-ids:iface-id=13d2cb29-d61c-46d9-
↪afe9-98b6aa0a43ea', 'external-ids:iface-status=active', u'external-ids:attached-
↪mac=fa:16:3e:6d:ac:ea', \
      'external-ids:vm-uuid=1065e38c-e6c6-423f-8140-5d0c021d3af0'].
↪Exception: Unexpected error while running command.
2016-01-27 11:37:22.289 12945 ERROR nova.compute.manager [req-b7fdc659-2fd5-4d9e-942c-
↪803f71c2cce1 d82509fae77e41009880defd0bbd829e d9c0a5bd157947bab06d355bf4772db7 - - -
↪] \
  [instance: 1065e38c-e6c6-423f-8140-5d0c021d3af0] Instance failed to spawn
2016-01-27 11:37:22.289 12945 ERROR nova.compute.manager [instance: 1065e38c-e6c6-423f-
↪8140-5d0c021d3af0] AgentError: Error during following call to agent: \
  ['ovs-vsctl', '--timeout=120', '--', '--if-exists', 'del-port', u'tap13d2cb29-d6', '-
↪-', 'add-port', 'br-int', u'tap13d2cb29-d6', '--', 'set', 'Interface', u'tap13d2cb29-
↪d6', \
      u'external-ids:iface-id=13d2cb29-d61c-46d9-afe9-98b6aa0a43ea', 'external-ids:iface-
↪status=active', u'external-ids:attached-mac=fa:16:3e:6d:ac:ea', \
      'external-ids:vm-uuid=1065e38c-e6c6-423f-8140-5d0c021d3af0']
2016-01-27 11:37:22.289 12945 ERROR nova.compute.manager [instance: 1065e38c-e6c6-423f-
↪8140-5d0c021d3af0]
```

There are interesting files regarding running instances available on the compute nodes, in `/var/lib/nova/instances`:

```
# tree /var/lib/nova/instances
/var/lib/nova/instances
|-- 54fff47b-fa5e-4401-8309-e2da66c01d66
|   |-- console.log
|   |-- disk
|   |-- disk.info
|   +-- libvirt.xml
|-- 7fb5ce27-cb7a-4a3b-94d8-afb84ddd3c5b
```

(continues on next page)

(continued from previous page)

```

| |-- console.log
| |-- disk
| |-- disk.info
| +-- libvirt.xml
|-- _base
| +-- 775fa67e40ab15538f2f01969e50a38078c09e9b
|-- compute_nodes
+-- locks
    |-- nova-775fa67e40ab15538f2f01969e50a38078c09e9b
    +-- nova-storage-registry-lock

```

For instance, you can find the libvirt domain file used to boot the VM:

```

# head /var/lib/nova/instances/54fff47b-fa5e-4401-8309-e2da66c01d66/libvirt.xml
<domain type="kvm">
  <uuid>54fff47b-fa5e-4401-8309-e2da66c01d66</uuid>
  <name>instance-00000003</name>
  <memory>524288</memory>
  <memoryBacking>
    <hugepages>
      <page size="2048" nodeset="0" unit="KiB"/>
    </hugepages>
  </memoryBacking>
  <numatune>

```

Or you can look at a currently running Nova instance console logs:

```

# tail /var/lib/nova/instances/54fff47b-fa5e-4401-8309-e2da66c01d66/console.log
ec2: #####
-----BEGIN SSH HOST KEY KEYS-----
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYT[... ]0sfj0fFcxJvE2Roc= root@compute1-vm
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQ[... ]JUyLnAaNv8oNz1AId root@compute1-vm
-----END SSH HOST KEY KEYS-----
Cloud-init v. 0.7.5 finished at Wed, 27 Jan 2016 12:37:49 +0000. Datasource_
↳ DataSourceEc2.
Up 18.67 seconds

Ubuntu 14.04.2 LTS vm1-compute1-vm-ubuntu-14 ttyS0

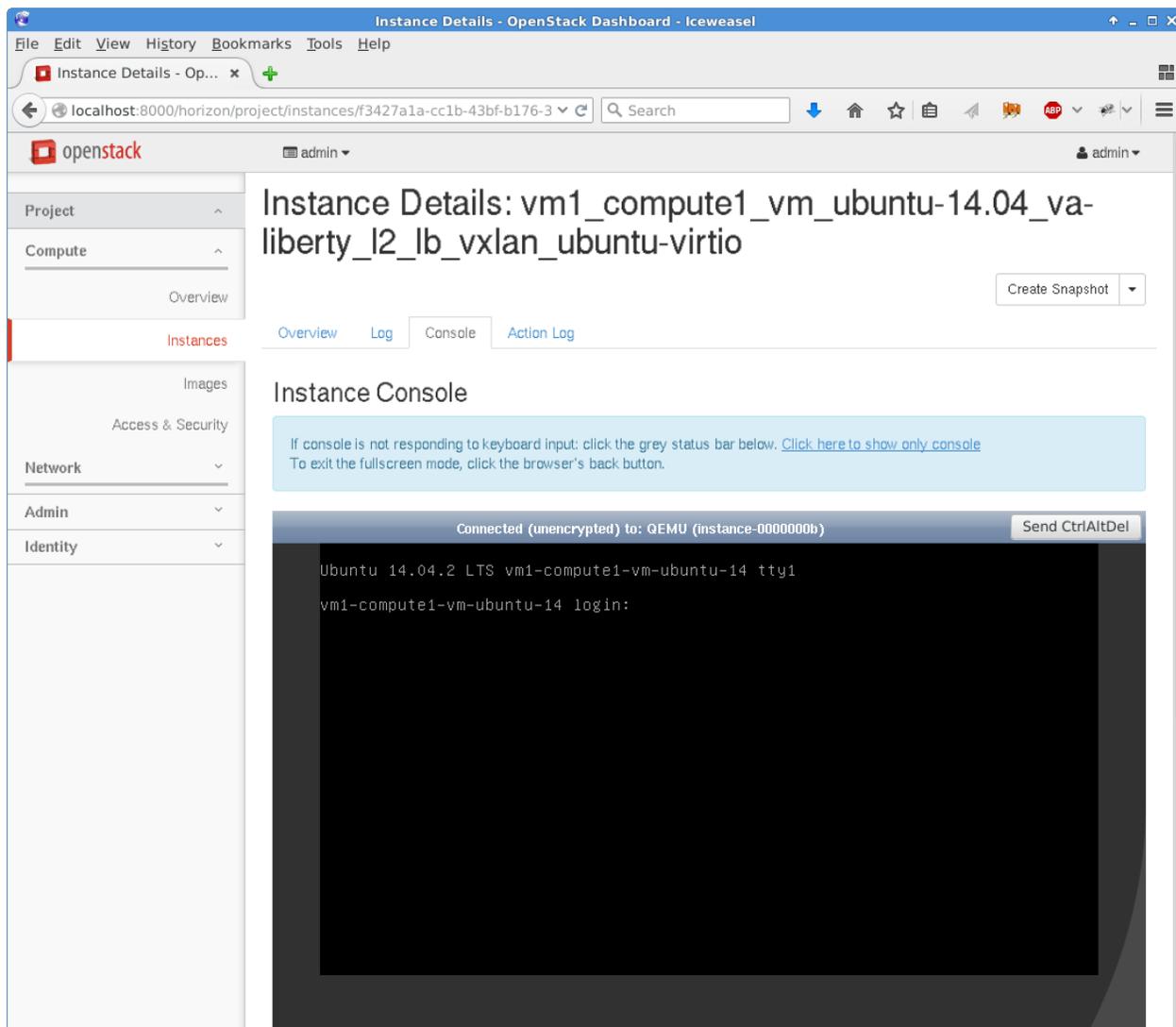
```

#### Note:

- For console logs, however, we recommend using `nova console-log <ID>` on the controller node, for a similar result.
- You may also access your VM console itself by accessing horizon (which must be installed and started obvi-

ously).

From the administration panel, access **Compute > Instances > [your instance] > Console**:



If you want Nova to provide you with more information when running, you can configure the `verbose` and `debug` options to `True` in `/etc/nova/nova.conf`:

```
# grep -iE "(verbose|debug)" /etc/nova/nova.conf
verbose = True
debug = True
```

Once configured, restart the `nova-compute` service.

- On Ubuntu Server:

```
# service nova-compute restart
```

- On Red Hat 7:

```
# systemctl restart openstack-nova-compute.service
```

Similarly, if you want Neutron to provide you with more information, configure verbose and debug options in `/etc/neutron/neutron.conf`:

```
# grep -iE "(verbose|debug)" /etc/neutron/neutron.conf
verbose = True
debug = True
```

Once configured, restart the Neutron service.

---

**Note:** Do not keep verbose and debug options set on production environments, as it is very, very talkative. It makes researching interesting information in the log difficult.

---

## 4.6 External Tools

### 4.6.1 strace

`strace` displays system calls done by a given program. Use this command to get a first impression on what the program is spending time on. For instance, you can see netlink messages handled by the cache manager:

```
# strace -p $(pidof cmgrd)
Process 5350 attached
setsockopt(11, SOL_SOCKET, SO_SNDBUF, [32768], 4) = 0
setsockopt(11, SOL_SOCKET, SO_RCVBUF, [32768], 4) = 0
bind(11, {sa_family=AF_NETLINK, pid=-2076175130, groups=00000000}, 12) = 0
getsockname(11, {sa_family=AF_NETLINK, pid=-2076175130, groups=00000000}, [12]) = 0
sendmsg(11, {msg_name(12)={sa_family=AF_NETLINK, pid=0, groups=00000000}, msg_iov(1)=[{
↪ "\34\0\0\0\20\0\5\0\204\315jV\3
6\24@\204\3\1\0\0\10\0\2\0vrf\0", 28}], msg_controllen=0, msg_flags=0}, 0) = 28
recvmsg(11, {msg_name(12)={sa_family=AF_NETLINK, pid=0, groups=00000000}, msg_iov(1)=[{
↪ "\320\0\0\0\20\0\0\0\204\315jV\
46\24@\204\1\2\0\0\10\0\2\0vrf\0\6\0\1\0"... , 16384}], msg_controllen=0, msg_flags=0}, ↪
↪ 0) = 208
recvmsg(11, {msg_name(12)={sa_family=AF_NETLINK, pid=0, groups=00000000}, msg_iov(1)=[{
↪ "$\0\0\0\2\0\0\0\204\315jV\346\
4@\204\0\0\0\0\34\0\0\0\20\0\5\0\204\315jV"... , 16384}], msg_controllen=0, msg_flags=0}
↪ , 0) = 36
```

(continues on next page)

(continued from previous page)

```

sendmsg(11, {msg_name(12)={sa_family=AF_NETLINK, pid=0, groups=00000000}, msg_iov(1)=[{
↪ "\24\0\0\0\33\0\5\3\205\315jV\3
6\24@\204\1\0\0\0", 20}], msg_controllen=0, msg_flags=0}, 0) = 20
recvmsg(11, {msg_name(12)={sa_family=AF_NETLINK, pid=0, groups=00000000}, msg_iov(1)=[{
↪ ",\0\0\0\33\0\2\0\205\315jV\346
24@\204\2\1\0\0\10\0\1\0\0\0\0\0\r\0\2\0"... , 16384}], msg_controllen=0, msg_flags=0}, ↵
↪ 0) = 44
epoll_wait(4,
^C
Process 5350 detached
<detached ...>

```

---

**Note:** Refer to the `strace` manpage for specific options.

---