Turbo IPsec

Release 2.2.22

6WIND

Jul 16, 2021

Contents

1. Overview

Thank you for choosing 6WIND Turbo IPsec.

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

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

Turbo IPsec 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 IPsec 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
- ECMP

- PBR (Policy-Based Routing)
- MPLS (Multiprotocol Label Switching) LDP (Label Distribution Protocol) (beta)
- BGP L3VPN (Layer 3 Virtual Private Network) (beta)
- BGP Flowspec

1.1.2 LAYER 2 (Data Link Layer) and Encapsulations

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

1.1.3 IP Networking

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

1.1.4 IPsec

- IKE (Internet Key Exchange)v1, IKEv2
- Encryption: 3DES, AES-CBC/GCM (128, 192, 256)
- Hash: MD-5, SHA-1, SHA-2 (256, 384, 512), AES-XCBC (128)
- RSA, Diffie-Helman Key Management
- High performance (AES-NI, QAT)
- Tunnel, Transport or BEET mode

1.1.5 Security

- Access Control Lists
- Unicast Reverse Path Forwarding

1.1.6 QoS

- Rate limiting per interface, per VRF

1.1.7 IP Services

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

1.1.8 Management/Monitoring

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

1.1.9 System

- Control Plane Protection

1.1.10 High Availability

- VRRP (Virtual Router Redundancy Protocol)
- IKE/IPsec synchronization

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 IPsec requires at least 2GB of RAM. Default capabilities are automatically adjusted to the amount of RAM available.

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

VRs (Virtual Routers)	32
Routes	1000000
Neighbors	100000
PBR rules	4096
Netfilter rules	10000
Netfilter conntracks	262144
Netfilter ebtables	10000
Netfilter ipset	64 ipsets per VR (Virtual Router), 2048 entries per ipset
VXLAN interfaces	512
IPsec tunnels	100000

See also:

Fast path limits configuration to tune these capabilities.

- CPU: Turbo IPsec requires at least 2 CPU cores.
- Storage: Turbo IPsec 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 IPsec.

2.1 Delivery contents

The Turbo IPsec delivery contains:

- `6wind-turbo-ipsec-ee-...-doc.tgz`

  Turbo IPsec documentation, including:

  - main product documentation (this document)
  - list of publicly available software included in the Turbo IPsec binaries
  - description files for supported SNMP MIBs (Management Information Bases)

- `bin/`

  Turbo IPsec images in various formats, as described in the *Installation* section.

- `.md5` files to check integrity of deliverables

2.2 Installation

2.2.1 Install on bare metal using USB stick

This chapter explains how to try Turbo IPsec 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 IPsec* without changing anything on your machine
- *Install Turbo IPsec* 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-turbo-*-<arch>-<version>.img.gz
# dd if=6wind-turbo-*-<arch>-<version>.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 IPsec

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 IPsec.

> **Warning:** Please make sure that there is no other Turbo IPsec 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~beta2-36ubuntu3.11

+----------------------------------------------------------------------------+
|*Turbo IPsec - 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.

```
(...)
[  OK  ] Started 6WIND Turbo Appliances.
(...)
localhost login:
```

Note: Please contact 6WIND support if you do not get the login, or if you get a different status than OK for 6WIND Turbo Appliances in the logs.

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 IPsec

Once you have tried Turbo IPsec, 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 IPsec 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 IPsec 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 IPsec 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.

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~beta2-36ubuntu3.11

+----------------------------------------------------------------------------+
|*Turbo IPsec - 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.

(...)
[  OK  ] Started 6WIND Turbo Appliances.
(...)
localhost 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 IPsec 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 IPsec* without changing anything on your machine
- *Install Turbo IPsec* on a local disk

Test Turbo IPsec

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 IPsec.

> **Warning:** Please make sure that there is no other Turbo IPsec 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~beta2-36ubuntu3.11

+----------------------------------------------------------------------------+
|*Turbo IPsec - 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.

```
(...)
[  OK  ] Started 6WIND Turbo Appliances.
(...)
localhost login:
```

Note: Please contact 6WIND support if you do not get the login, or if you get a different status than OK for 6WIND Turbo Appliances in the logs.

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 IPsec

Once you have tried Turbo IPsec, 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 IPsec 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 IPsec 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`.

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

This command will install Turbo IPsec on `/dev/sda`. The relevant configuration files will be copied from the CDROM drive to the local drive. At the end of the installation, you can reboot and unload the CDROM.

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~beta2-36ubuntu3.11

+----------------------------------------------------------------------------+
|*Turbo IPsec - 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.

(...)
[  OK  ] Started 6WIND Turbo Appliances.
(...)
localhost 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 IPsec 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 IPsec `iso` file and requires a deployment infrastructure enabling PXE by providing DHCP, TFTP, DNS and HTTP services.

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
```

(continues on next page)

(continued from previous page)

```
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 IPsec deliverables and the PXE server together to finalize the PXE infrastructure.

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

Kernel and filesystem of the installer:

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

Turbo IPsec ISO image:

```
# cp 6wind-turbo-*.iso $HTTP_DIR/turbo.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 IPsec 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␣
↪nonetworking 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.iso␣
↪ds=nocloud-net;s=http://$pxe_default_server/cloud-init/"
        echo "Boot options: $boot_opts"
        echo "Loading kernel image $kernel_image ..."
        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␣
↪AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBNqR+NMuQUywXp5+uqSc6WSFjxLRpRZoA9b7ekB

runcmd:
 - '/usr/bin/wget "http://192.168.235.1/cloud-init/vrouter.startup" -O /etc/
↪sysrepo/data/vrouter.startup'
 - '/usr/bin/install.sh -r -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 IPsec 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+uqSc6WSFjxLRpRZoA9b7ekB
↪"
            ]
          }
```

(continues on next page)

(continued from previous page)

```
        ]
      }
    },
    "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
        }
      }
    ]
  }
}
EOF
```

This startup configuration:

- sets host1 as the hostname,
- disables Turbo IPsec 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 IPsec.

Deploy Turbo IPsec on the target

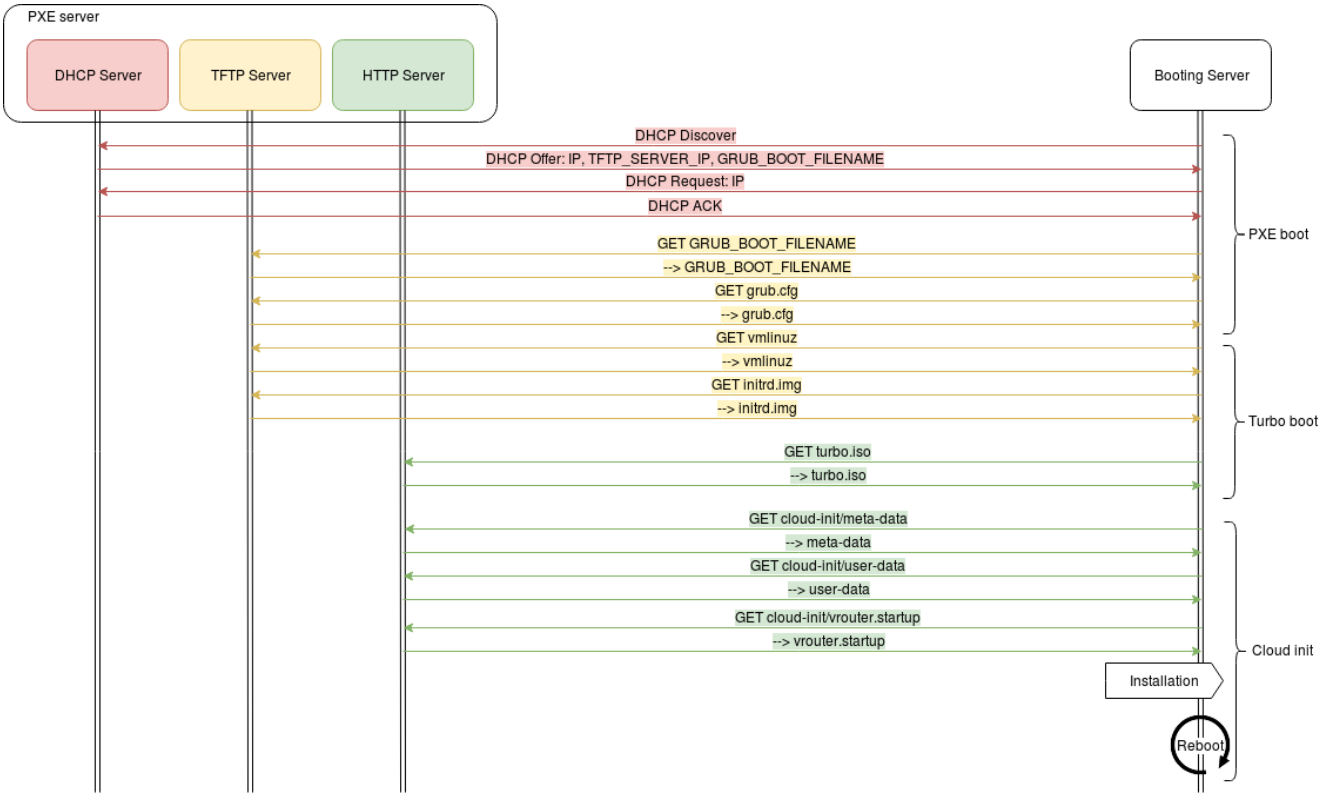
You target must be configured to boot in Legacy BIOS mode (UEFI is not supported) and first on the hard drive selected for installation.

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 IPsec installer kernel and initrd, using the Turbo IPsec `iso` as root filesystem
- execute the cloud-init script to:
  - configure the root account (no password, SSH key)
  - install Turbo IPsec 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 IPsec.

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 IPsec 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 libvirtd
```

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 IPsec in libvirt as a guest VM, use:

```
# cp turbo-ipsec-ee.qcow2 /var/lib/libvirt/images/vm1.qcow2
# virt-install --name vm1 --vcpus=3,sockets=1,cores=3,threads=1 \
               --os-type linux --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
(...)
Login:
```

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

VM with physical NICs

This section details how to start Turbo IPsec 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 IPsec, see *Passthrough mode*, simpler configuration, but only one VM can use each NIC
- portions of NICs are dedicated to Turbo IPsec, 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 IPsec 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 IPsec 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 IPsec 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 IPsec 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 IPsec 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)
```

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-ipsec-ee.qcow2 /var/lib/libvirt/images/vm1.qcow2
# virt-install --name vm1 --vcpus=3,sockets=1,cores=3,threads=1 \
               --os-type linux --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
(...)
Login:
```

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 IPsec 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
```

2. 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
```

(continues on next page)

(continued from previous page)

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

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

4. 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 IPsec.

```
# cp turbo-ipsec-ee.qcow2 /var/lib/libvirt/images/vm1.qcow2
# virt-install --name vm1 --vcpus=3,sockets=1,cores=3,threads=1 \
               --os-type linux --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
```

5. Connect to the VM:

```
# virsh console vm1
(...)
Login:
```

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 IPsec 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 IPsec 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
          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
```

(continues on next page)

(continued from previous page)

```
      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 IPsec 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 IPsec 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)
      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 IPsec 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 IPsec 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 IPsec 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 IPsec `qcow2` in OpenStack, creates the right flavor, and starts a Turbo IPsec VM.

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

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

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

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

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

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

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 IPsec VM with one interface on each network:

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

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

```
(...)
Login:
```

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

VM with physical NICs

This section details how to start Turbo IPsec 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 IPsec, see *Passthrough mode*, simpler configuration, but only one VM can use each NIC
- portions of NICs are dedicated to Turbo IPsec, 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 IPsec 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 IPsec 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=eno1
# 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 IPsec `qcow2` file path:

```
# TURBO_QCOW2=/path/to/6wind-turbo-*-<arch>-<version>.qcow2
# IFACE=eno1
# 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 IPsec `qcow2` file:

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

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

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

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

```
# openstack flavor set turbo-ipsec-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-ipsec-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 IPsec VM:

```
# openstack server create --flavor turbo-ipsec-passthrough \
                          --image turbo-ipsec \
                          turbo-ipsec_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 IPsec 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 IPsec 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" }'
```

(continues on next page)

(continued from previous page)

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 IPsec `qcow2` file path:

```
# TURBO_QCOW2=/path/to/6wind-turbo-*-<arch>-<version>.qcow2
# IFACE=eno1
# 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 IPsec `qcow2` file:

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

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

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

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

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

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

```
# openstack server create --flavor turbo-ipsec-sriov \
                          --image turbo-ipsec \
                          turbo-ipsec_vm
```

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

2.2.6 Install as a VM using VMware

VMware basic deployment

Turbo IPsec 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=200
and that one (https://kb.vmware.com/selfservice/search.do?cmd=displayKC&docType=kc&docTypeID=DT_KB_1_1&external
for compatibility. Turbo IPsec'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 IPsec will not be able to use them.

In order to boot your Turbo IPsec 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)

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 IPsec): 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  %OVRLP   %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
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
```

(continues on next page)

(continued from previous page)

```
      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
 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
```

(continues on next page)

(continued from previous page)

```
 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 IPsec VM using Proxmox VE and the `.iso` file.

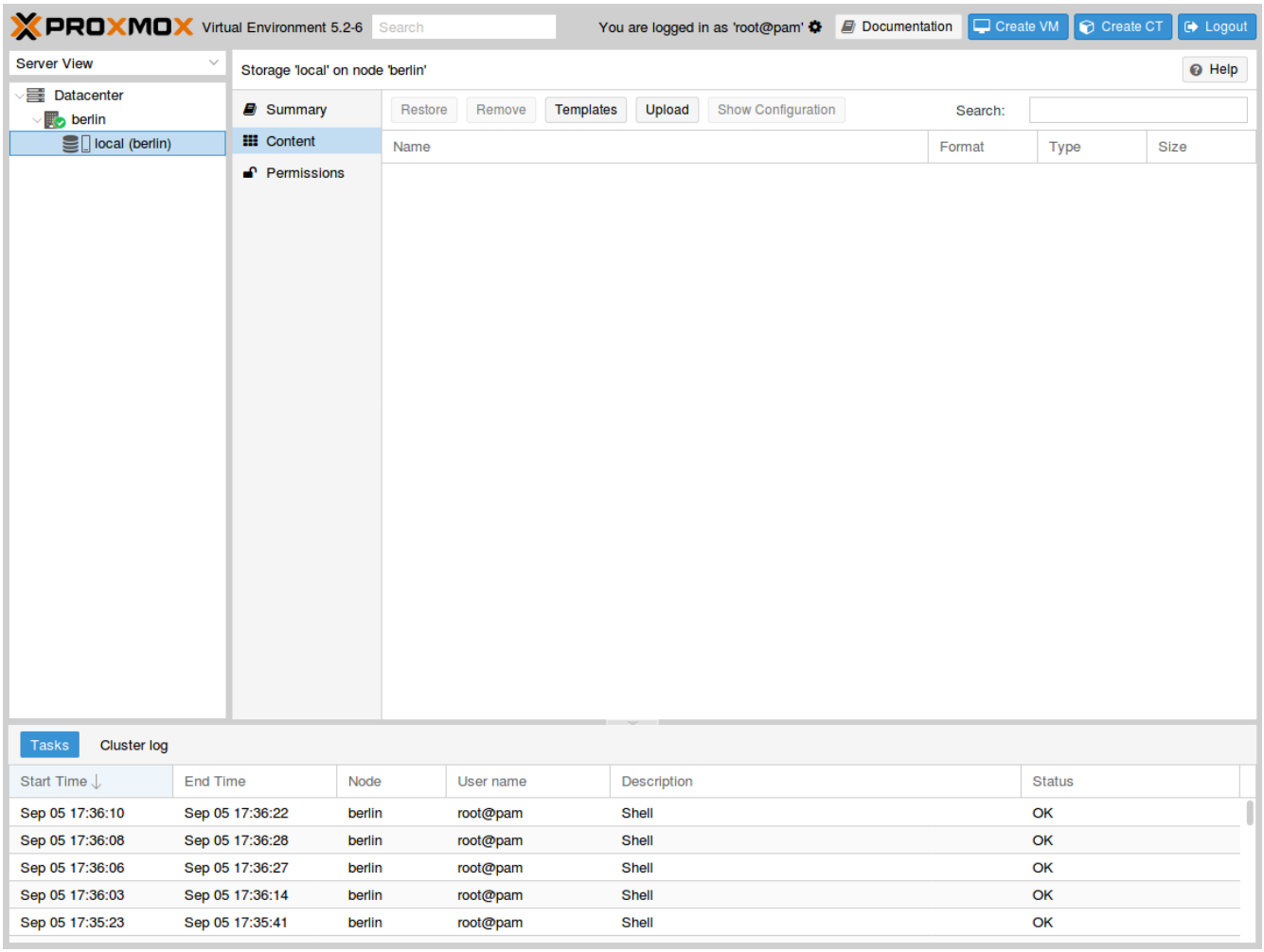
It expects that you already installed a Proxmox VE cluster, in which you are able to spawn VMs with network connected.

It follows the following steps:

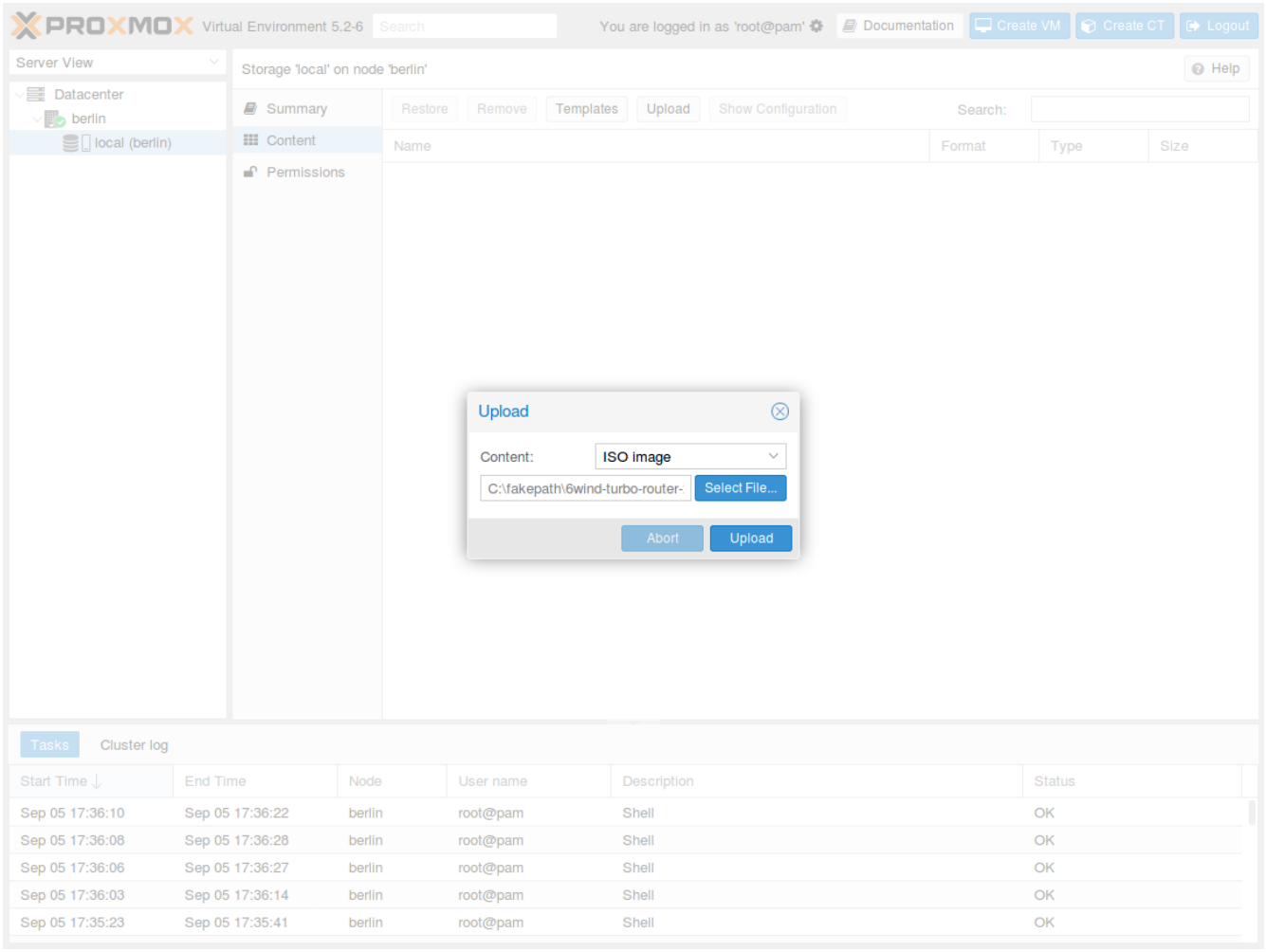
- make the `.iso` file available to Proxmox VE
- create and configure a VM
- boot the VM using the `.iso` file
- install Turbo IPsec on the virtual disk

Upload the `.iso` file

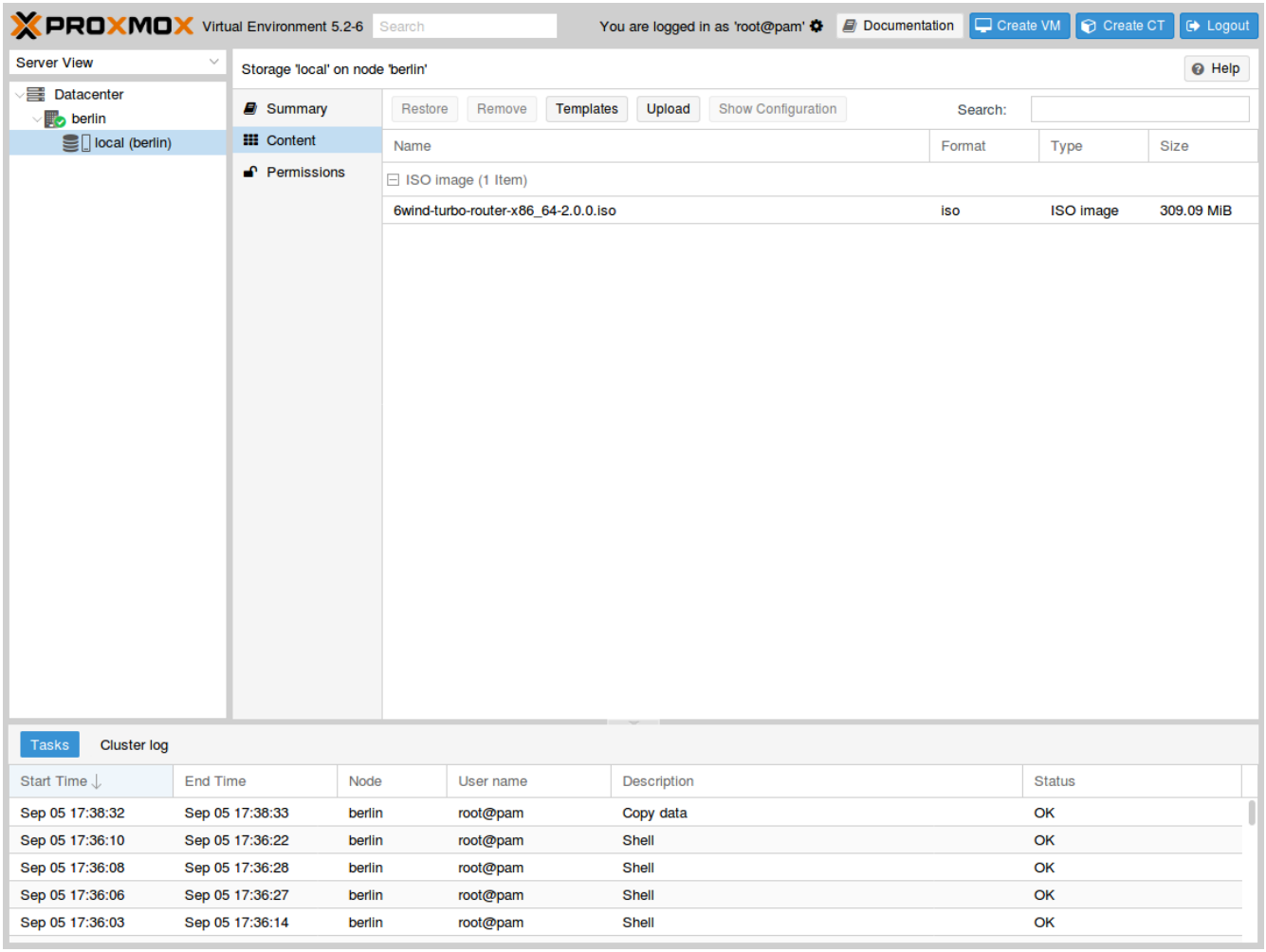
Select the local storage of your node in the left pane and visualize its content:



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

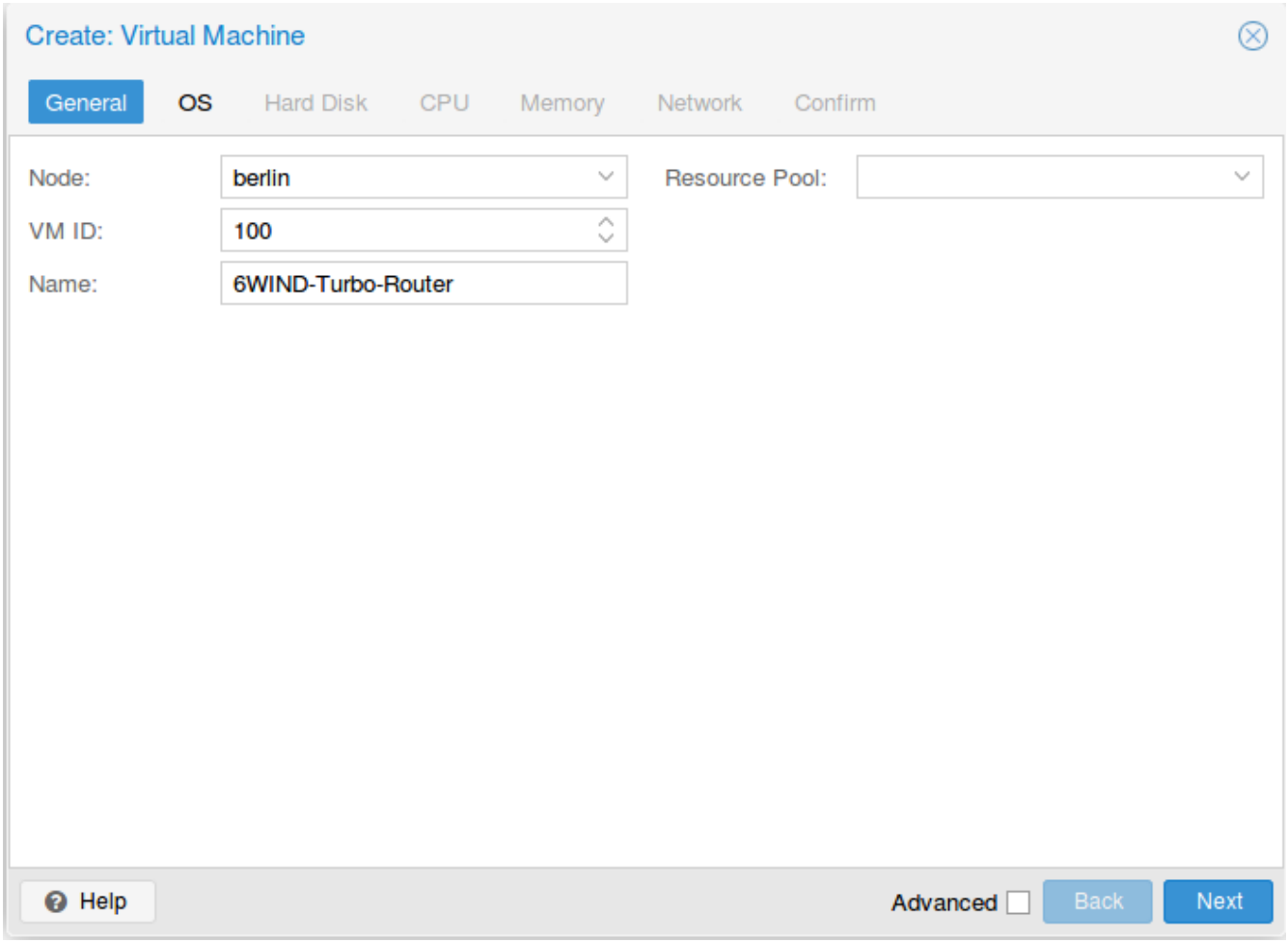


The `.iso` file is now available to this node:

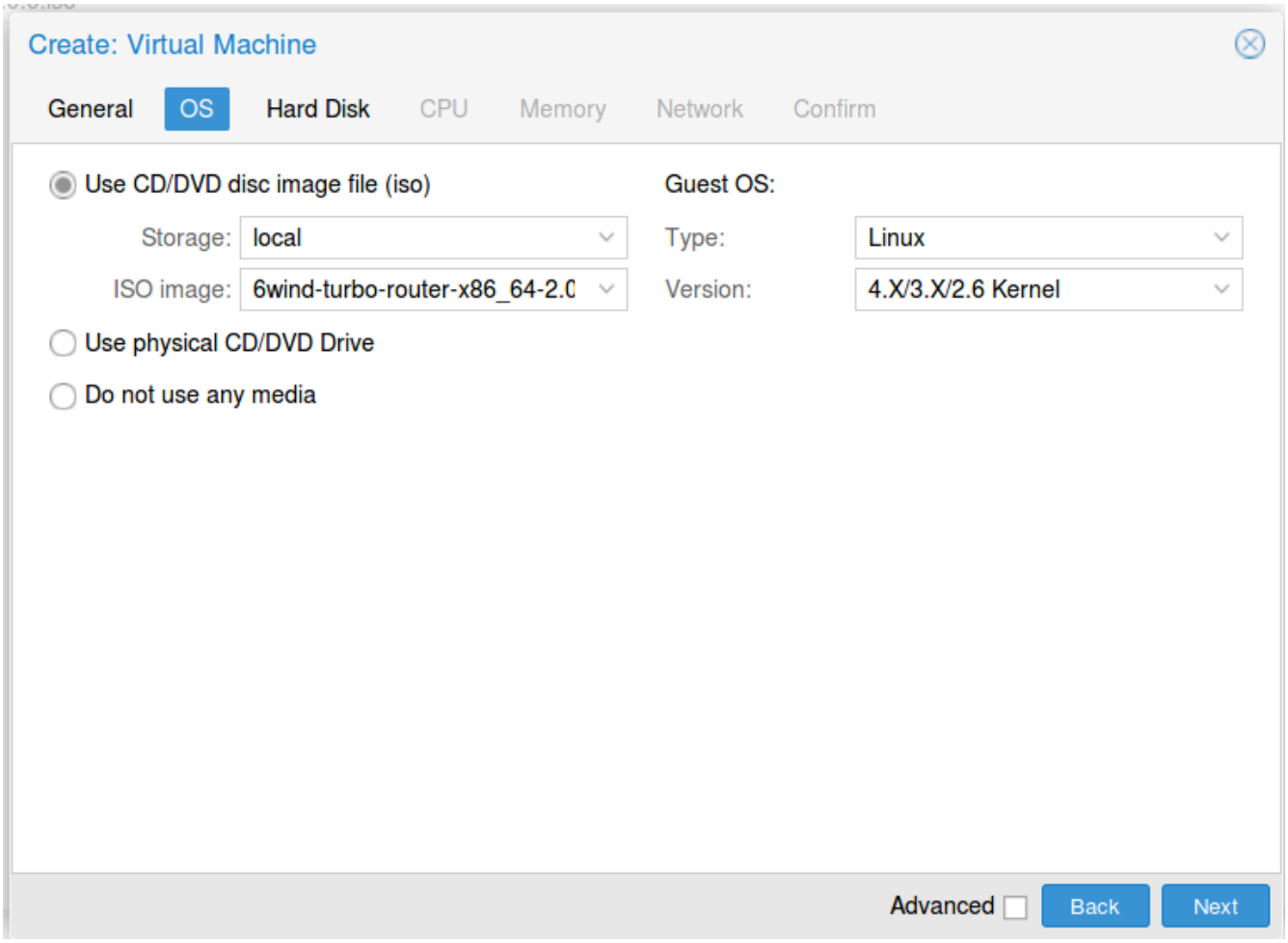


Create and boot the VM

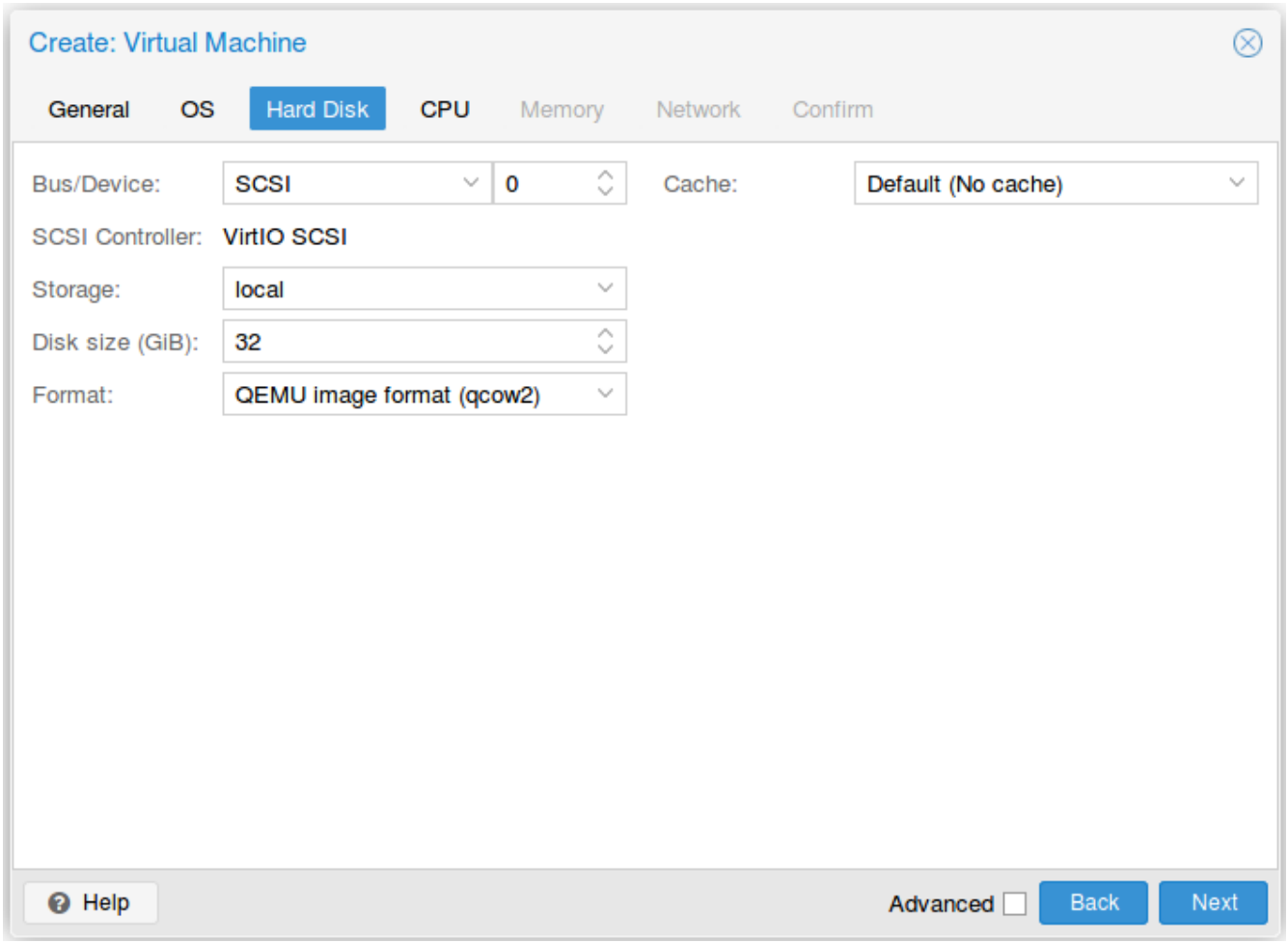
In the top right corner, press the `Create VM` button to launch the creation wizard. In `General` tab, check the node and the VM ID, and give a name to the VM, then press `Next`:



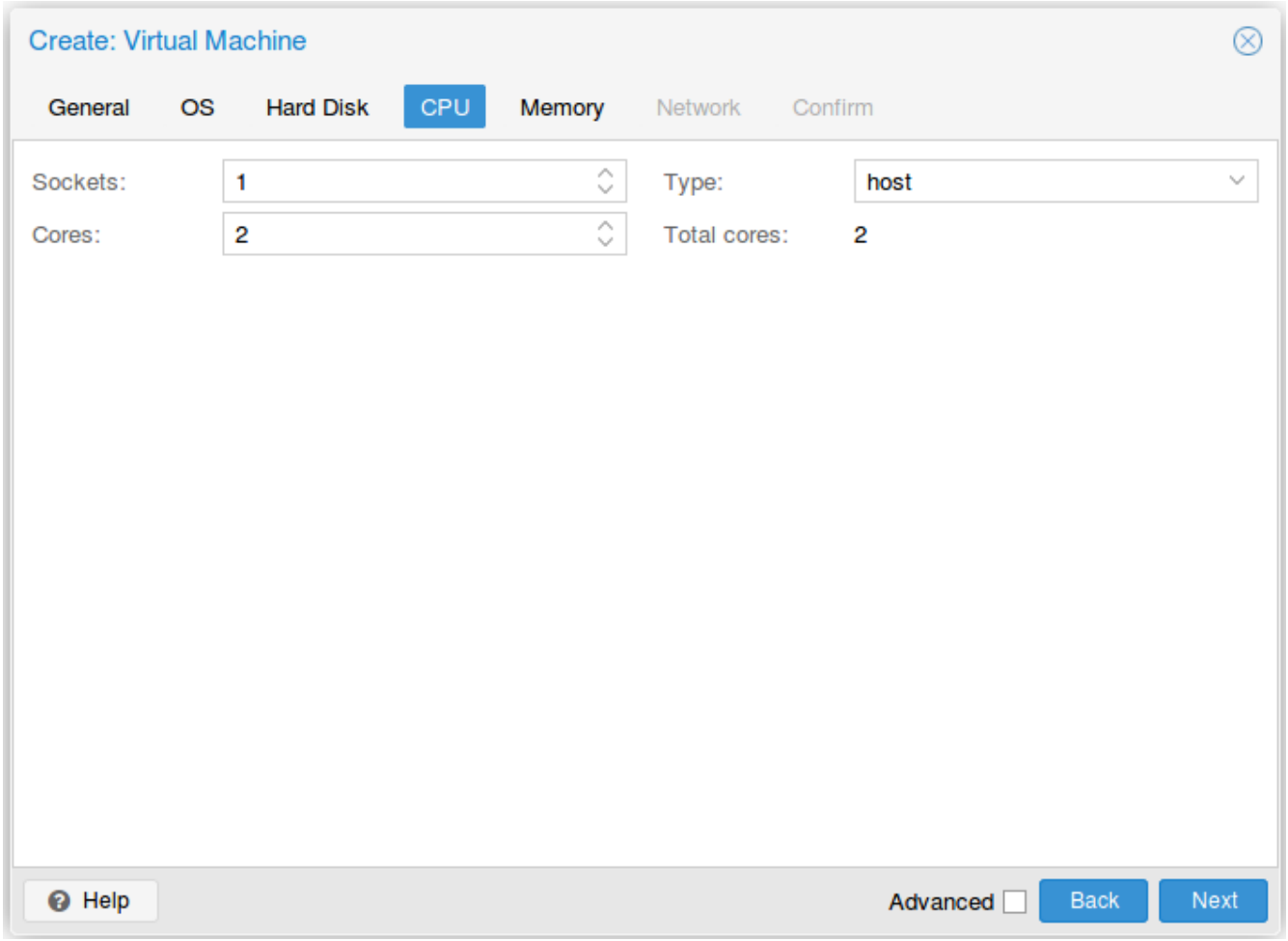
In `OS` tab, make sure to use the uploaded `.iso` file as CD/DVD and to specify a Linux with 4.X/3.X/2.X kernel as `Guest OS`, then press `Next`:



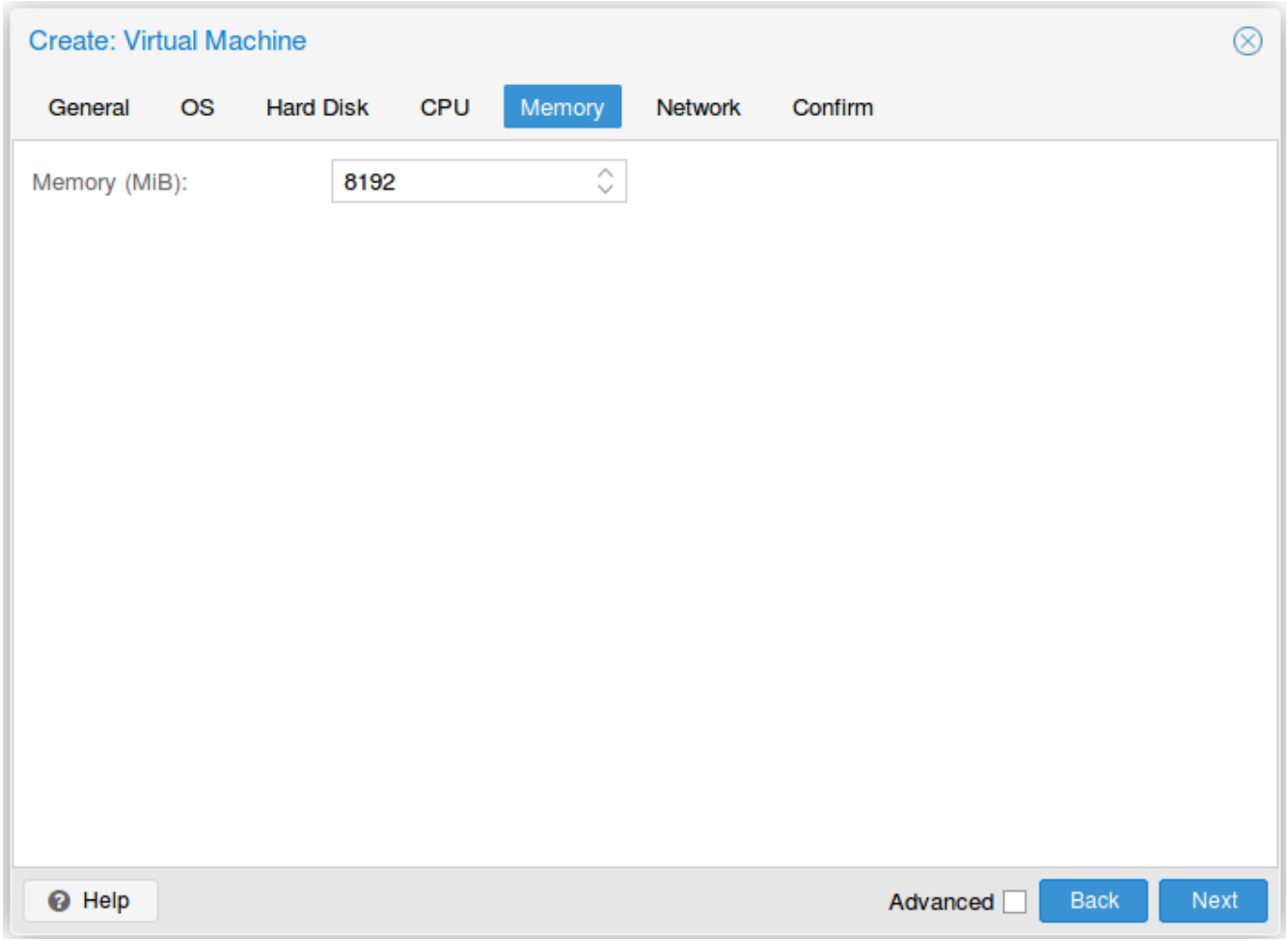
In `Hard Disk` tab, keep the default qcow2 device with VirtIO SCSI storage and allocate at least 10GB, then press `Next`:



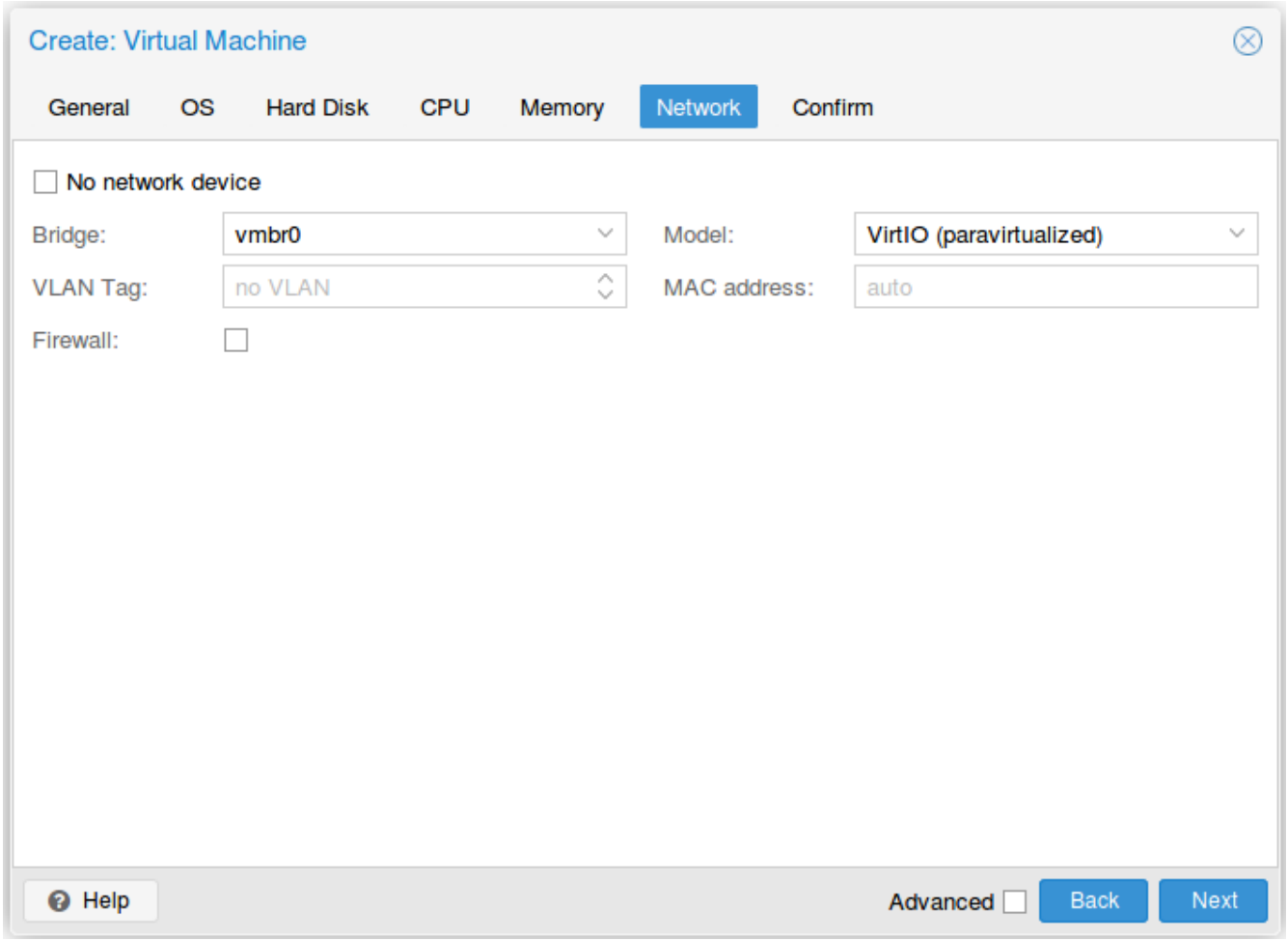
In `CPU` tab, allocate at least 2 cores and select `host` as CPU type, then press `Next`:



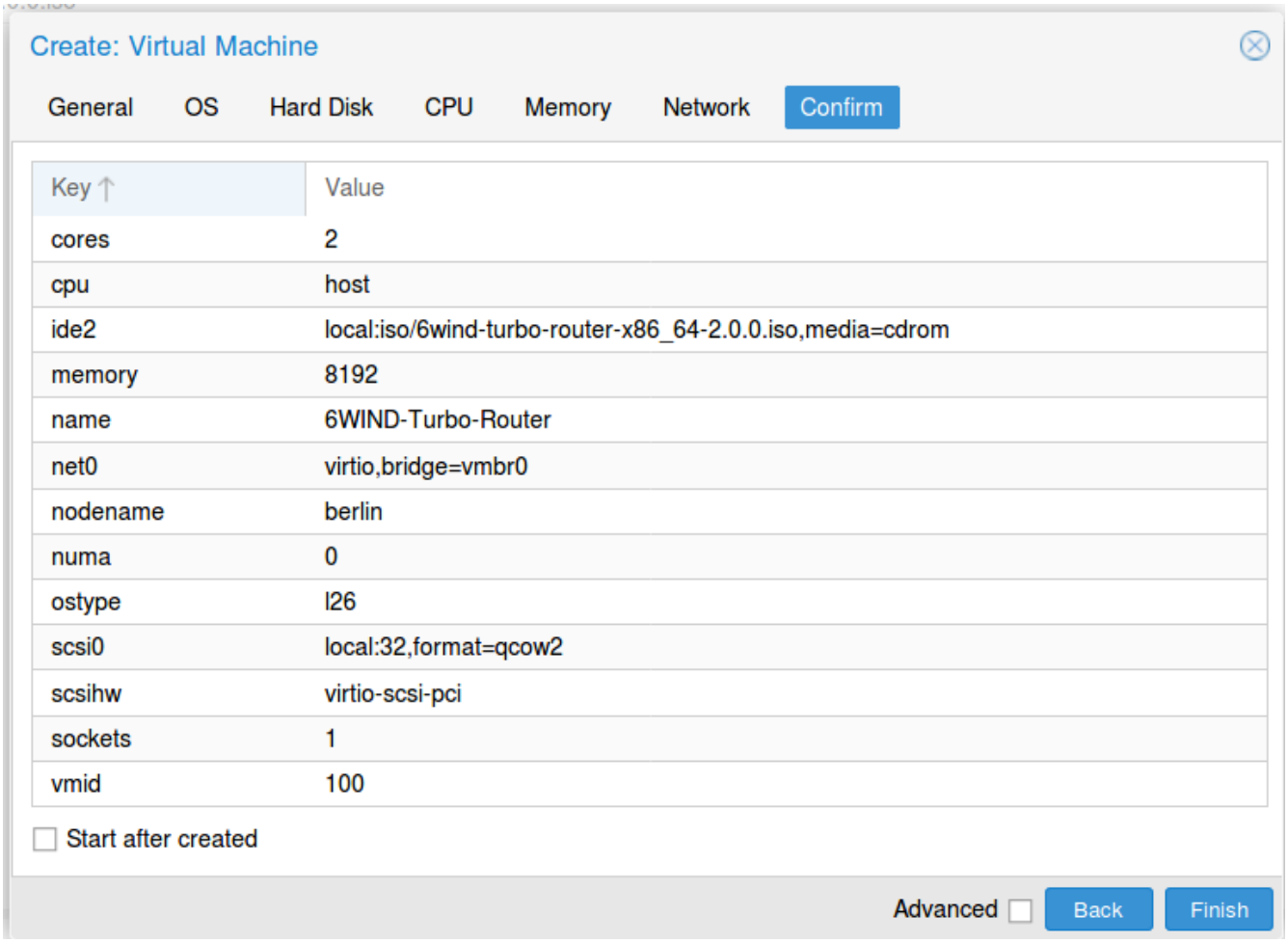
In `Memory` tab, allocate at least 8GB of RAM, then press `Next`:



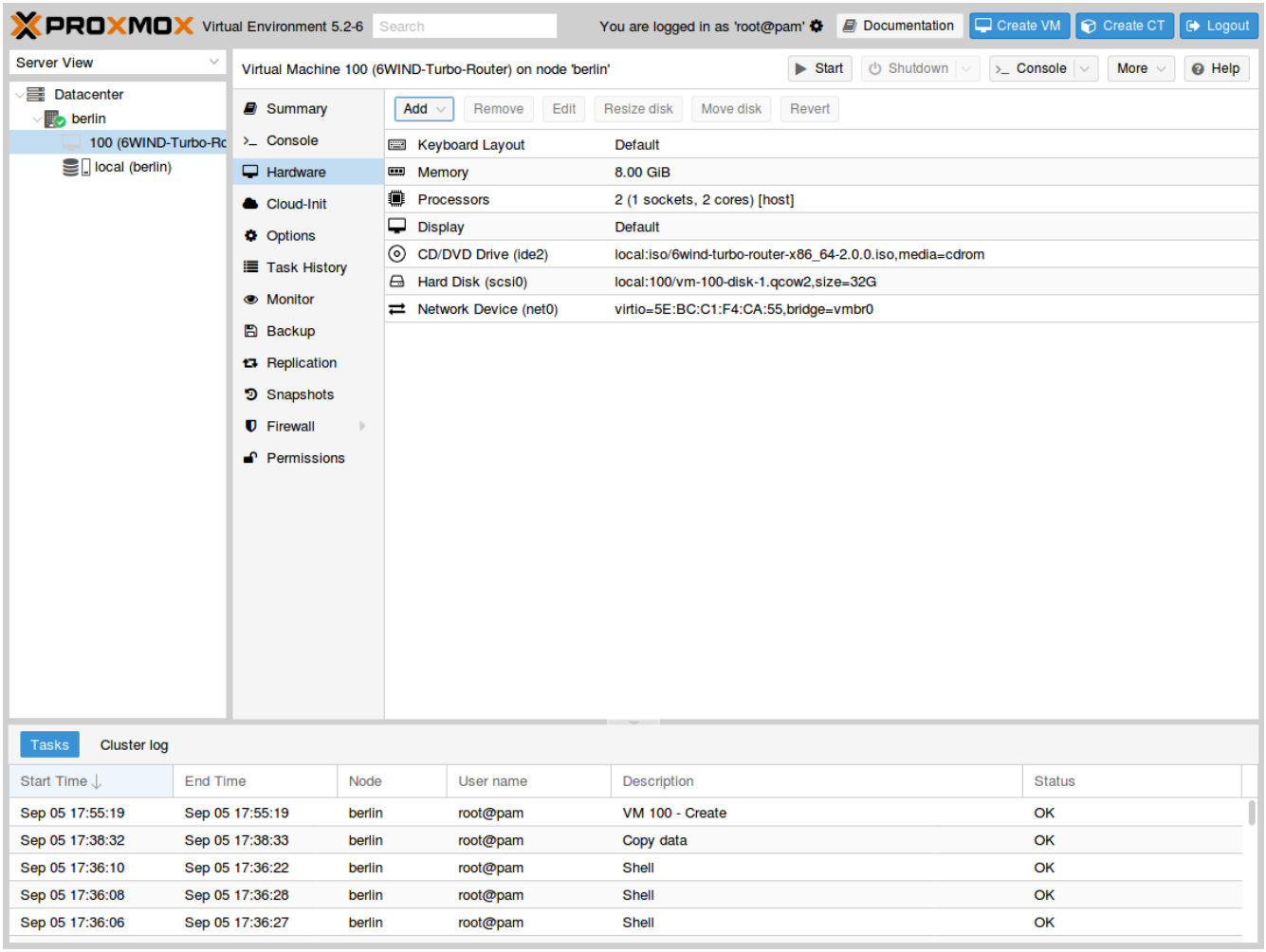
In `Network` tab, bind the virtual management interface to a host bridge in order to have access to external network. Select `VirtIO` as model type, then press `Next`:



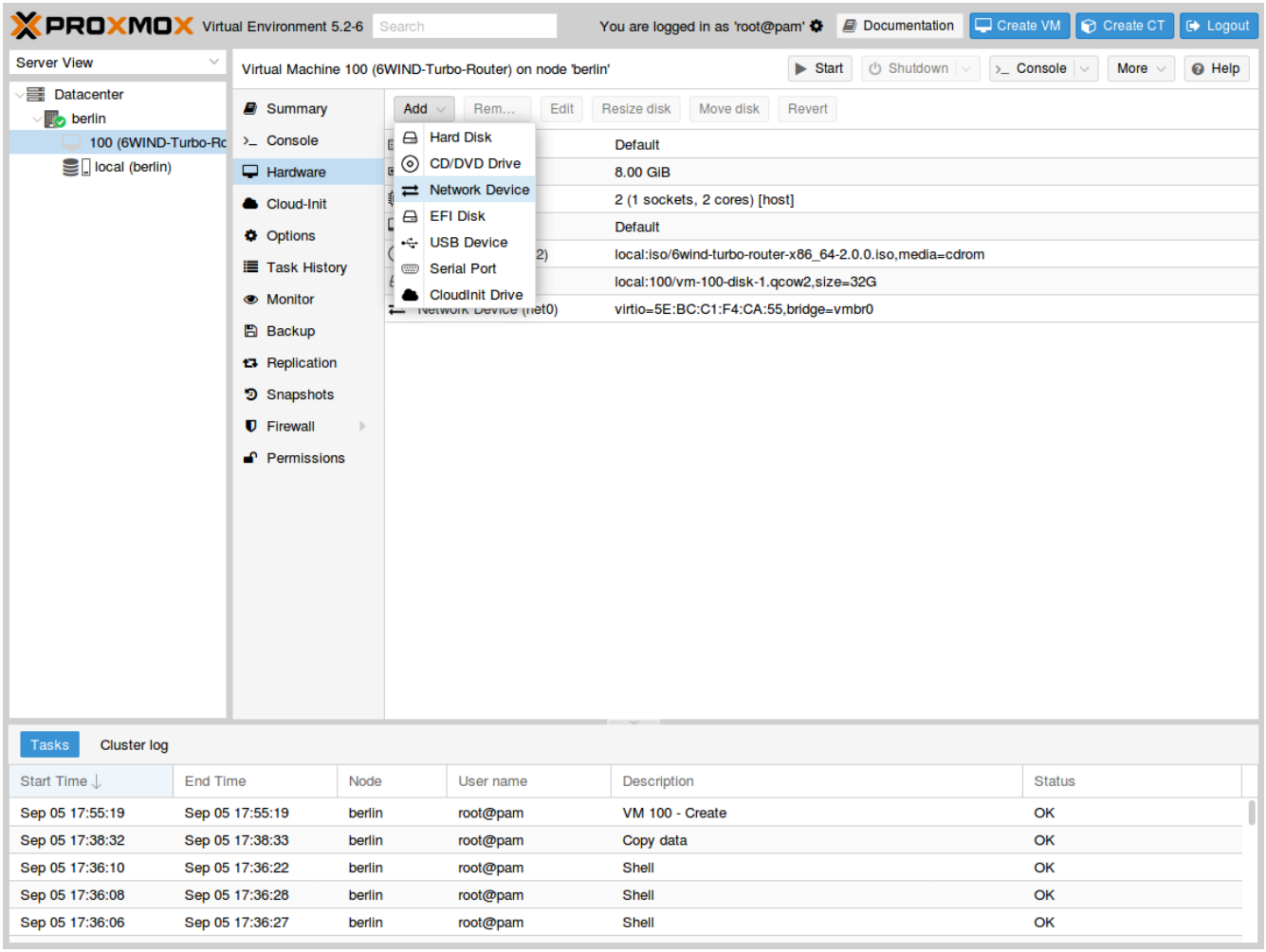
In `Confirm` tab, review your settings and press `Finish` to finalize the creation and get back to the main dashboard:



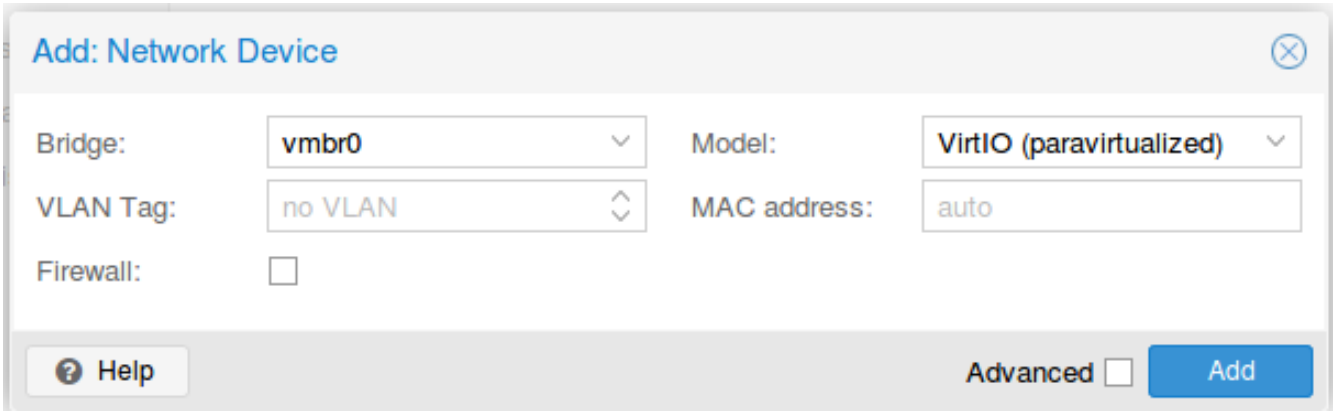
The VM is now available in the left pane below your physical node. Select it and review its hardware configuration:



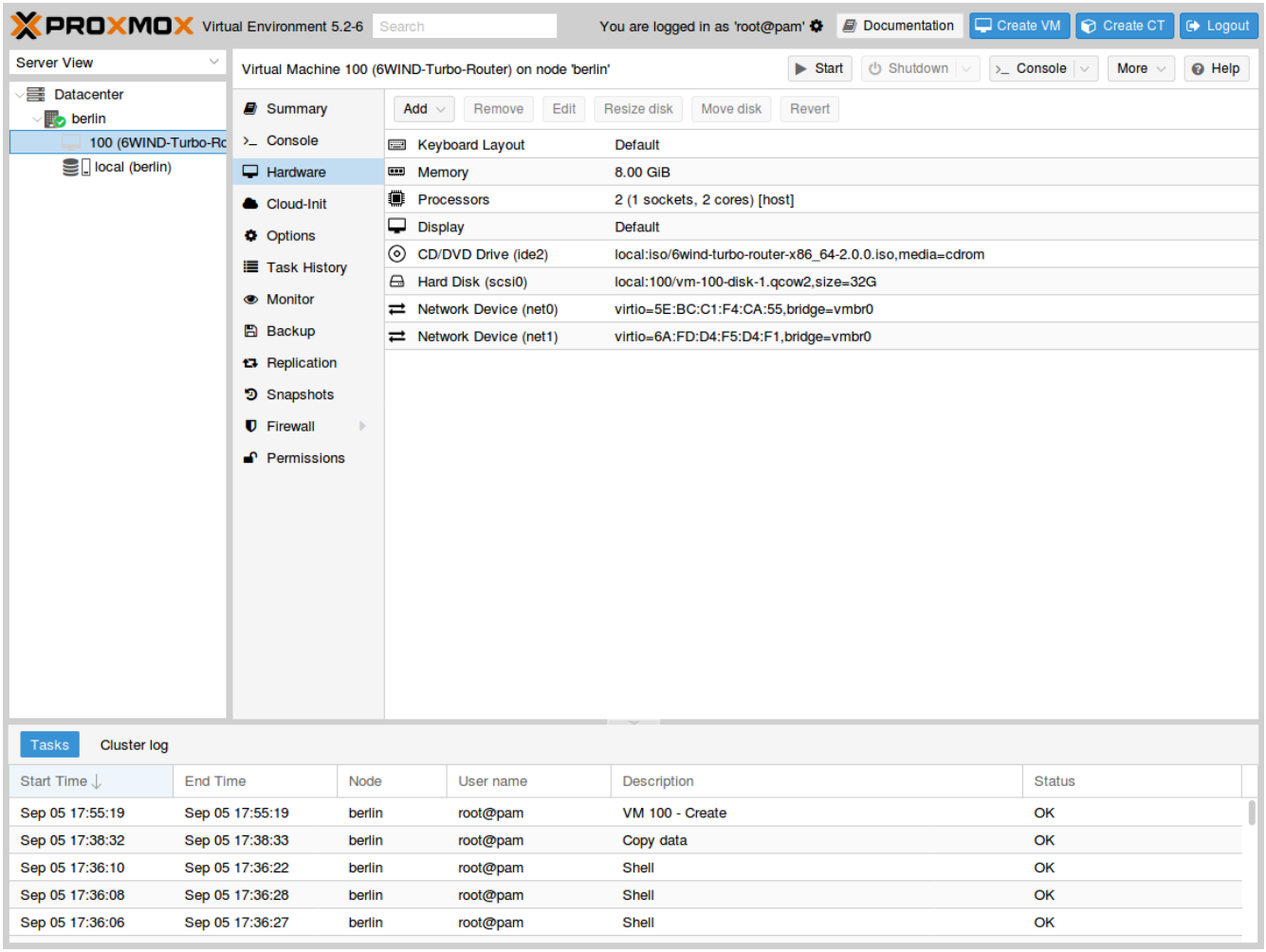
Press `Add` > `Network Device`:



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:



Warning: Please make sure that there is no other Turbo IPsec 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 IPsec

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 IPsec on `/dev/sda`. The relevant configuration files will be copied to the local drive.

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

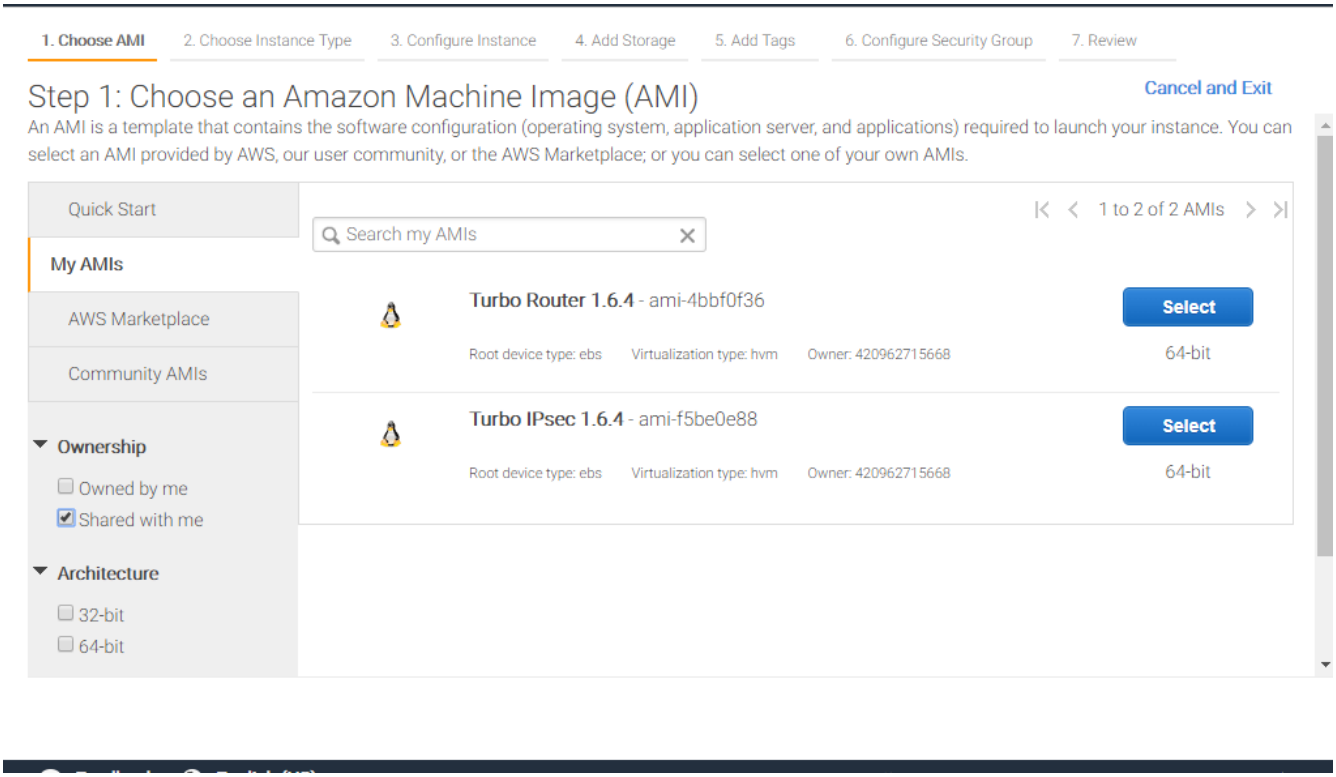
Once access is granted, the Turbo IPsec AMI will be available in the AWS management console when selecting `AMIs > Private Images`.

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`.

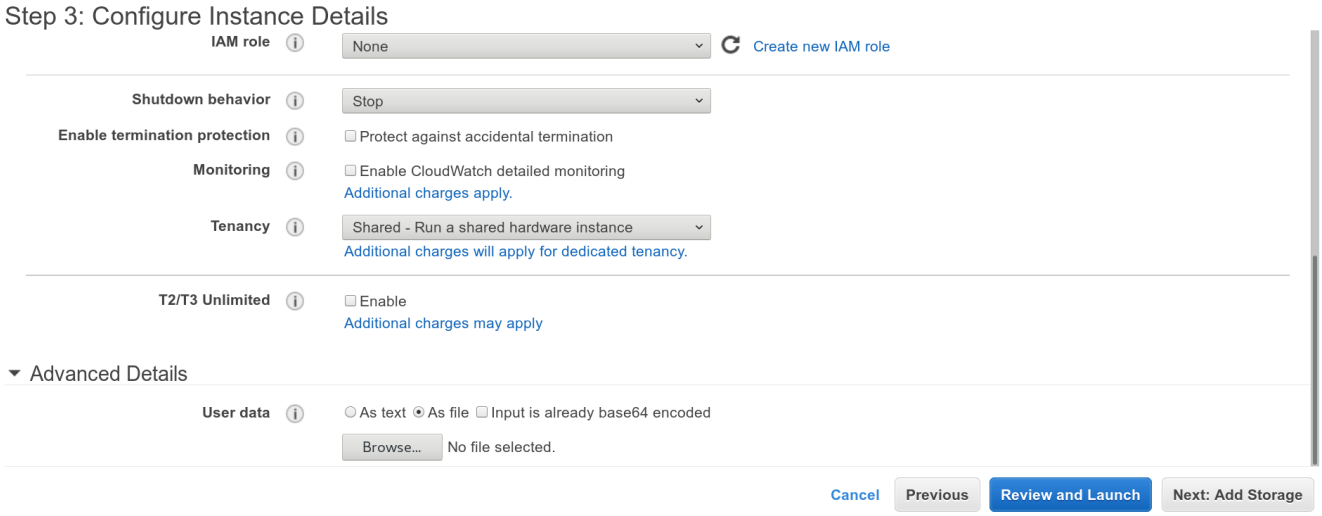


Step 2: choose instance type

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`.



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 IPsec 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 lturbol component in charge of packet processing.) and enabling VLAN stripping.

```
#cloud-config
write_files:
- path: /etc/turbo.lic
  content: |
   LICENSE 6wind turbo-router 01.99.99 permanent uncounted
     hostid=isv=628CE7A75DA9EFB7B3A2D3CDEB566889 customer=yourcompany
     _ck=c082fce984 sig="60PG4527MCR2KEKTD2UP7TRN18G1R6GDJCUM2XH508A03PHQ
     BQ168E3GWWK3VQ43TK0YPQ01KWVG"
- path: /etc/sysrepo/data/vrouter.startup
   content: |
      {
         "vrouter:config": {
            "vrf": [
               {
                  "name": "main",
                  "vrouter-interface:interface": {
                     "physical": [
                        {
                           "name": "pub1",
                           "port": "pci-b0s5",
                           "ipv4": {
```

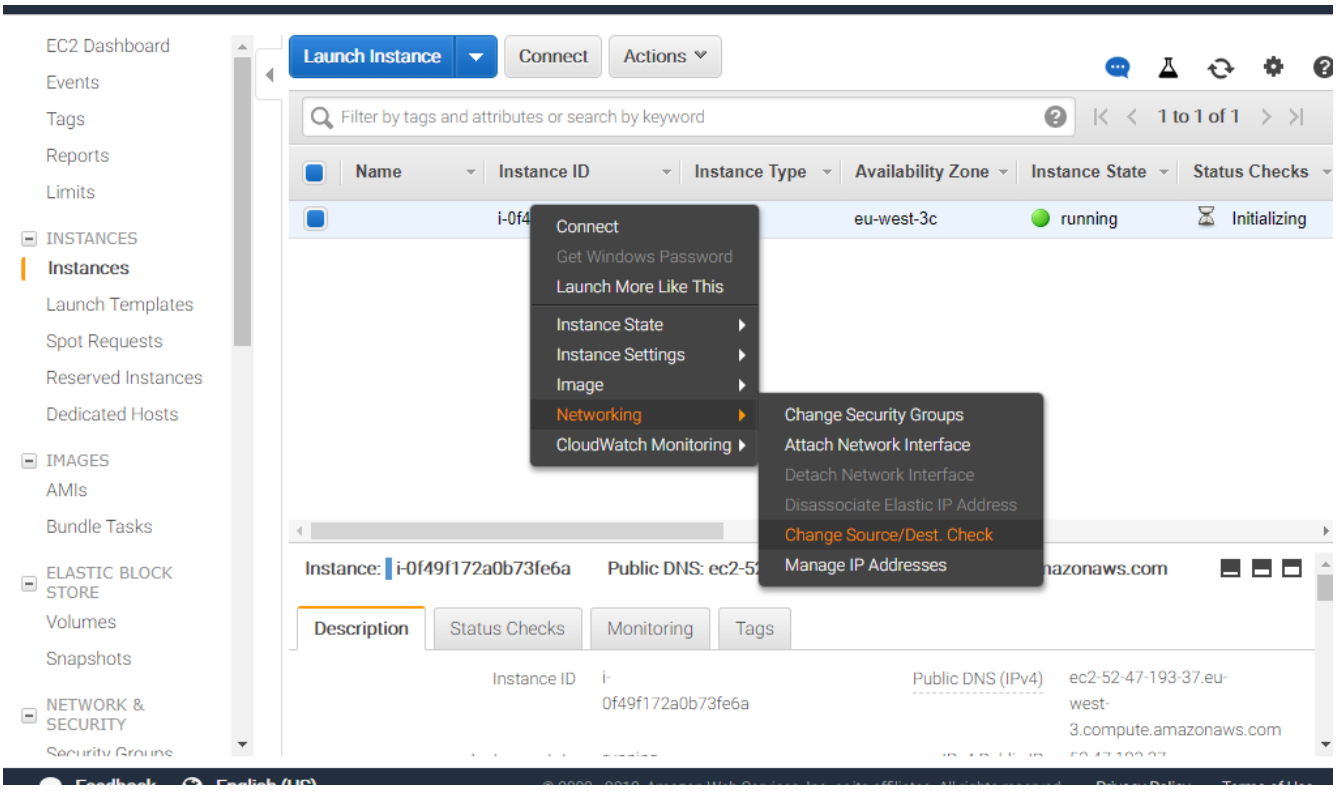
(continues on next page)

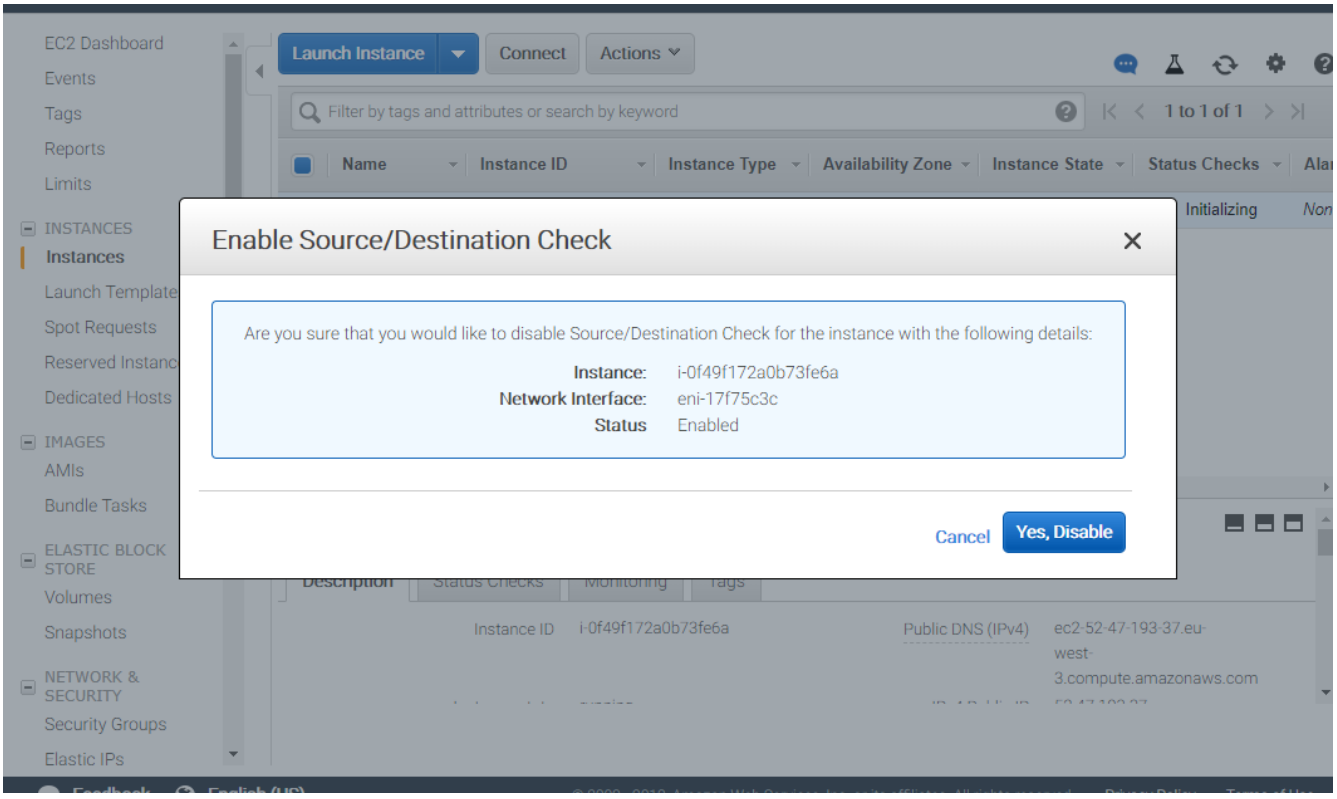
(continued from previous page)

```
                        "dhcp": {
                           "enabled": true
                         }
                      }
                   }
                ]
             }
          }
       ],
       "vrouter-system:system": {
          "vrouter-fast-path:fast-path": {
             "port": [
                "pci-b0s5"
             ],
             "advanced": {
                "vlan-strip": true
             }
          }
       }
    }
}
```

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*.

This section explains how to install a Turbo IPsec appliance.

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

Method	Flavor
Install on bare metal using USB stick	`img.gz`
Install on bare metal using CDROM or *using PXE*	`iso`
Install as a VM using KVM	`qcow2`
Install as a VM using VMware	`ova`
Install as a VM using AWS	`ami`[1]
Update an existing installation[2]	`update`

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

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 Day-1 configuration

Automatic Day-1 configuration

Turbo IPsec 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
    (...)
```

(continues on next page)

[1] Contact your customer support representative to get access to a private `ami`.
[2] Turbo IPsec 2.x only. To update from Turbo IPsec 1.x, a fresh installation is required.

(continued from previous page)

```
    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
```

(continues on next page)

(continued from previous page)

```
        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 equipement 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 and default route via DHCP 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. 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
            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. Enable the SSH server:

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

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

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

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

2.3.3 Installing your license file

If no license file is present, Turbo IPsec will start and run for 48 hours. After this period, its functionality and performance will be degraded. If you do not wish to install a license file now, you may jump to the next section.

Your license file will be provided to you by the 6WIND support team.

Use the `license` command to import and install your license file.

```
vrouter> cmd license import scp://user:password@ip//absolute/path/to/file.
↪lic
License download queued.
```

(continues on next page)

(continued from previous page)

```
vrouter> cmd license status
Download success (size 207).
```

Note: The import command accepts other url types, like ftp and http. Run `cmd license import <?>` for a complete list. Also, the server integrity is not checked for scp, sftp and https.

Note: The license validity is checked when the fast path is started. Run `cmd license status` after the fast path has been started for more information.

2.3.4 Configuring the fast path

The fast path is the Turbo IPsec 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
```


2.3.5 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 IPsec 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 IPsec.

See also:

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

Customizing the Turbo IPsec 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 and upload a license file.

```
# cat << EOF > /tmp/user-data
#cloud-config
write_files:
 - path: /etc/turbo.lic
   content: |
     LICENSE 6wind turbo-ipsec 01.99.99 permanent uncounted
       hostid=isv=628CE7A75DA9EFB7B3A2D3CDEB566889 customer=yourcompany
       _ck=c082fce984 sig="60PG4527MCR2KEKTD2UP7TRN18G1R6GDJCUM2XH508A03PHQ
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-type linux --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 and upload a license file.

```
# cat << EOF > /tmp/user-data
#cloud-config
write_files:
 - path: /etc/turbo.lic
   content: |
     LICENSE 6wind turbo-ipsec 01.99.99 permanent uncounted
       hostid=isv=628CE7A75DA9EFB7B3A2D3CDEB566889 customer=yourcompany
```

(continues on next page)

(continued from previous page)

```
      _ck=c082fce984 sig="60PG4527MCR2KEKTD2UP7TRN18G1R6GDJCUM2XH508A03PHQ
chpasswd:
  list: |
    root:myrootpassword
EOF
```

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

```
# openstack server create --flavor turbo-ipsec \
                          --image turbo-ipsec \
                          --user-data /tmp/user-data \
                          turbo-ipsec_vm
```

Examples

Here is a `user-data` example, where we pre-install the license file (make sure you replace the contents by your own), and we upload a startup configuration for the CLI (you can also upload alternative configurations).

```
#cloud-config
write_files:
- path: /etc/turbo.lic
  content: |
   LICENSE 6wind turbo-router 01.99.99 permanent uncounted
     hostid=isv=628CE7A75DA9EFB7B3A2D3CDEB566889 customer=yourcompany
     _ck=c082fce984 sig="60PG4527MCR2KEKTD2UP7TRN18G1R6GDJCUM2XH508A03PHQ
     BQ168E3GWWK3VQ43TK0YPQ01KWVG"
- path: /etc/sysrepo/data/vrouter.startup
   content: |
      {
         "vrouter:config": {
            "vrf": [
               {
                  "name": "main",
                  "vrouter-interface:interface": {
                     "physical": [
                        {
                           "name": "pub1",
                           "port": "ens1",
                           "ipv4": {
                              "dhcp": {
                                 "enabled": true
                               }
                            }
                         }
                      ]
                   }
                }
             ],
```

(continues on next page)

(continued from previous page)

```
            "vrouter-system:system": {
               "vrouter-fast-path:fast-path": {
                  "port": [
                     "pci-b0s5"
                  ]
               }
            }
         }
      }
```

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 beeing 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:

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

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.

Getting help

The CLI provides a comprehensive help system, which can be invoked in different ways.

The command `help` displays a context-sensitive list of available commands. In edition mode, the general commands are displayed first, followed by the context specific commands.

```
vrouter running physical eth0# help
  cmd                   Send a command.
  commit                Commit configuration.
  copy                  Copy a configuration into another one.
  del                   Delete a configuration node.
  echo                  Echo arguments.
  exec                  Execute a cli script file.
  exit                  Quit the edition mode.
  export                Export a configuration file.
  help                  Show the help.
  import                Import a configuration file.
```

(continues on next page)

(continued from previous page)

```
  load                  Load a configuration in staging (overwrite current one).
  netconf               NETCONF related commands: connect, disconnect, status.
  pwd                   Show current path.
  remove                Remove a configuration file.
  resize                Resize terminal.
  save                  Save the current staging configuration in a file.
  show                  Show configuration or system state.
  validate              Validate current configuration.
  yang                  YANG related commands: Load, list, show.

  ..                    Go to parent.
  /                     Go to root.
  description           A textual description of the interface.
  enabled               The desired (administrative) state of the interface.
  ethernet              Top-level container for Ethernet configuration.
  ipv4                  Parameters for the IPv4 address family.
  ipv6                  Parameters for the IPv6 address family.
  mtu                   Set the max transmission unit size in octets.
  port                  Reference to a physical network port.
```

The `help` command is also used to display a more detailed help of a command:

```
== show ==
Show configuration or system state.
show config:
  Show the configuration.
  In edition mode, display the staging configuration.
  In operational mode, display the running configuration.
  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
    ..
network-port pci-b0s3
    pci-bus-addr 0000:00:03.0
    vendor "Red Hat, Inc."
    model "Virtio network device"
    mac-address de:ad:de:01:02:03
    ..
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
    ..
```

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 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:

```
vrouter> 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:

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

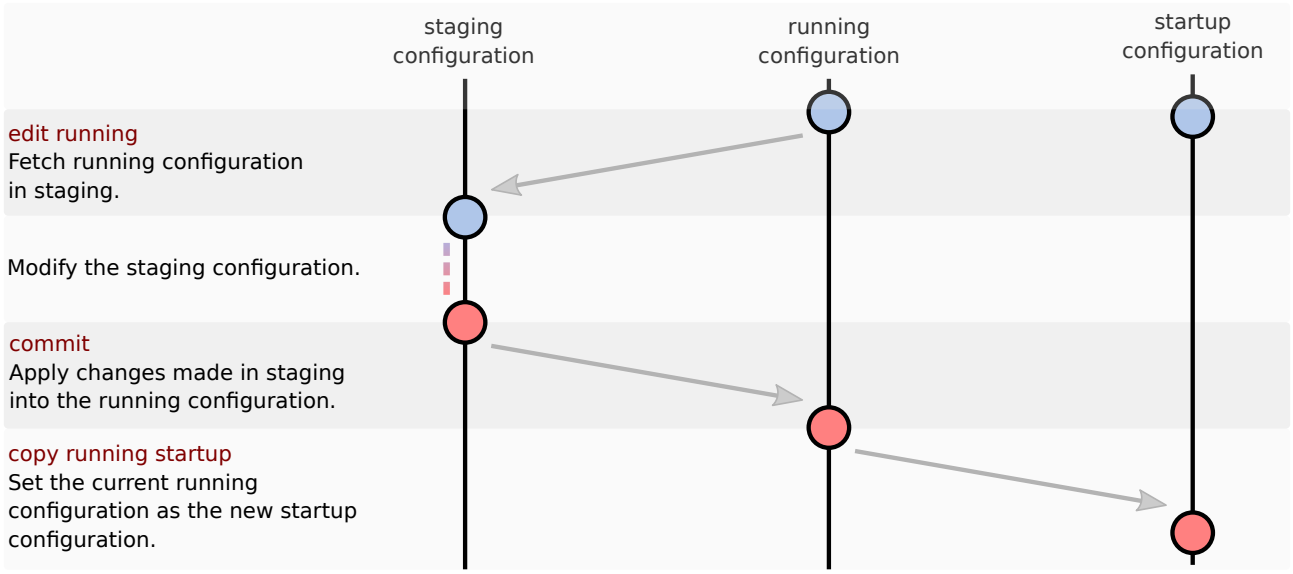
By default, the output is in text format. But it can also be displayed in `xml` or `json`.

Show the interfaces of the device in xml format:

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

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:

```
vrouter> edit running
vrouter 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
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 *oneliner* 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
```

(continues on next page)

(continued from previous page)

```
   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
         ..
      ..
   ..
```

(continues on next page)

(continued from previous page)

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

```
vrouter 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:

```
vrouter 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
            ..
        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
```

(continues on next page)

(continued from previous page)

```
-     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
+ / 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:

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

vrouter> 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:

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

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

3.1.4 System

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 `admin` role.
`viewer`	`viewer`	A restricted account for monitoring purposes. It has the `viewer` role.
`root`	`6windos`	Provides the ability to log into the Linux subsystem as superuser. Note that, unless documented otherwise, this account must not be used to configure the system, as it would conflict with the CLI.

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` acccount 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/QlMS5Anp7LTq$Lws2OmAm0SO.cBmPBGtdpwnfdAM4hDM4AdSO4ncXjS/
```

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/QlMS5Anp7LTq$Lws2OmAm0SO.cBmPBGtdpwnfdAM4hDM4AdSO4ncXjS/
```

`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>
        <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 IPsec 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:UrMHdqPxmoeV8DNYRtL0Il5cWAFfzZn7PHy4j2enH5A robobuild@ubuntu1604es
The key's randomart image is:
+---[ECDSA 256]---+
```

(continues on next page)

(continued from previous page)

```
|         .o+++..oo|
|          +o+ . oo|
|         O O o   .|
|        + ^ +  .. |
|       . 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/
↪1qkapobkS6yCnwauqTEveBw1GOjwuTADvqQVozBoaLbY3KGmsI= user@my-laptop
```

```
vrouter running user john# authorized-key "ecdsa-sha2-nistp256␣
↪AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBH2hK42JHtTYU1XRw2Zu4xCriM7CIXBl19p1/
↪1qkapobkS6yCnwauqTEveBw1GOjwuTADvqQVozBoaLbY3KGmsI= 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/
↪1qkapobkS6yCnwauqTEveBw1GOjwuTADvqQVozBoaLbY3KGmsI= user@my-laptop

root@vrouter:~# cat >> /root/.ssh/authorized_keys
ecdsa-sha2-nistp256␣
↪AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBH2hK42JHtTYU1XRw2Zu4xCriM7CIXBl19p1/
↪1qkapobkS6yCnwauqTEveBw1GOjwuTADvqQVozBoaLbY3KGmsI= 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 IPsec - 2.2

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 afer 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 IPsec 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"
   }
}

user = john {
     name = "John C"
     member = admins
     pap = PAM
}

user = alice {
```

(continues on next page)

(continued from previous page)

```
      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">
```

(continues on next page)

(continued from previous page)

```
  <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
        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">
  <system xmlns="urn:6wind:vrouter/system">
    <network-stack>
      <ipv4>
        <forwarding>true</forwarding>
        <send-redirects>true</send-redirects>
        <accept-redirects>true</accept-redirects>
```

(continues on next page)

(continued from previous page)

```
        <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:

```
vrouter running config# vrf vr1 network-stack ipv4
vrouter running ipv4# accept-redirects false
vrouter running ipv4# .. ipv6
vrouter running ipv6# accept-redirects false
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
        ..
```

(continues on next page)

(continued from previous page)

```
    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>
```

Neighbor

The maximum number of neighbors entries is limited.

To change these limits, do:

```
vrouter running config# system
vrouter running system# network-stack
vrouter running network-stack# neighbor
vrouter running neighbor# ipv4-max-entries 4096
vrouter running neighbor# ipv6-max-entries 4096
vrouter 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:

```
vrouter> 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:

```
vrouter 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# conntrack
vrouter running conntrack# max-entries 1000000
vrouter running conntrack# commit
```

Warning: If the fast path is running, a similar change is required in *fast path limits configuration*.

To customize conntrack TCP/UDP timeouts:

```
vrouter running config# system
vrouter running system# network-stack
vrouter running network-stack# conntrack
vrouter running conntrack# tcp-timeout-close 20
vrouter running conntrack# tcp-timeout-close-wait 70
vrouter running conntrack# tcp-timeout-established 500000
vrouter running conntrack# tcp-timeout-fin-wait 130
vrouter running conntrack# tcp-timeout-last-ack 40
vrouter running conntrack# tcp-timeout-max-retrans 400
vrouter running conntrack# tcp-timeout-syn-recv 70
```

(continues on next page)

(continued from previous page)

```
vrouter running conntrack# tcp-timeout-syn-sent 130
vrouter running conntrack# tcp-timeout-time-wait 130
vrouter running conntrack# tcp-timeout-unacknowledged 400
vrouter running conntrack# udp-timeout 40
vrouter running conntrack# udp-timeout-stream 190
vrouter running conntrack# commit
```

To display the conntrack state:

```
vrouter> show state / system network-stack conntrack
conntrack
    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
    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:

```
vrouter running conntrack# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <system xmlns="urn:6wind:vrouter/system">
    <network-stack>
      <conntrack>
        <max-entries>1000000</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>
      </conntrack>
    </network-stack>
  </system>
</config>
```


Fast path

The fast path is the Turbo IPsec 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:

```
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
    cp-protection
        budget 10
        ..
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.

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">
  <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
        ..
    limits
        fp-max-vrf 16
        ..
```

Note: fast path starting can take several seconds.

Configuring the core masks

In the `core-mask` context, the assignation 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">
  <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>
```

(continues on next page)

(continued from previous page)

```
    </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:

```
vrouter> edit running
vrouter running config# system fast-path
vrouter running fast-path#! port pci-b0s4
vrouter running fast-path# limits
vrouter running limits# fp-max-vrf 128
vrouter running limits# ip4-max-route 1000000
vrouter running limits# ..
vrouter 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
        ..
    ..
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">
```

(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/>
      <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, 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 (Internet Protocol Security) 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 criticity 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/>
      <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>
```

(continues on next page)

(continued from previous page)

```
        <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
  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.rx_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.

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">
```

(continues on next page)

(continued from previous page)

```
  <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.

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
Broadcast message from root@localhost (Fri 2018-07-13 14:59:13 CEST):

ATTENTION
The system is going down for reboot at Fri 2018-07-13 15:00:13 CEST!

poweroff-time "Fri 2018-07-13 15:00:13 CEST"
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@localhost (Fri 2018-07-13 14:59:19 CEST):

The system shutdown has been cancelled

OK.
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
Broadcast message from root@localhost (Fri 2018-07-13 14:59:13 CEST):

ATTENTION
The system is going down for poweroff at Fri 2018-07-13 15:00:13 CEST!

poweroff-time "Fri 2018-07-13 15:00:13 CEST"
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@localhost (Fri 2018-07-13 14:59:19 CEST):

The system shutdown has been cancelled

OK.
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
Turbo IPsec - 2.0.0 (default) (current)
vrouter>
```

One image is displayed per line in the following format:

```
<product name> - <version>[ - <image name>] [(default)] [(current)] [(next)]
```

<product name> 6WIND product name.

<version> 6WIND product version.

<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-turbo-ipsec-ee-x86_
↪64-2.0.1.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
Turbo IPsec - 2.0.0 (default) (current)
Turbo IPsec - 2.0.1 (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 2.0.1 new-name my-img
vrouter> cmd system-image list
Turbo IPsec - 2.0.0 (default) (current)
Turbo IPsec - 2.0.1 - my-img (next)
vrouter>
```

Change the default boot image

```
vrouter> cmd system-image set-default my-img
vrouter> cmd system-image list
Turbo IPsec - 2.0.0 (current)
Turbo IPsec - 2.0.1 - my-img (default) (next)
vrouter>
```

Remove an image:

```
vrouter> cmd system-image delete my-img
vrouter> cmd system-image list
Turbo IPsec - 2.0.0 (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>
```

(continues on next page)

(continued from previous page)

```
      </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
        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) 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
vrouter running remote-server 10.125.0.2# port 514
```

(continues on next page)

(continued from previous page)

```
vrouter running remote-server 10.125.0.2# log-filter facility any level greater-or-
↪equal error
vrouter running remote-server 10.125.0.2# log-filter facility kernel level any
vrouter running remote-server 10.125.0.2# log-filter facility auth level emergency
vrouter running remote-server 10.125.0.2# log-filter facility mail level none
vrouter running remote-server 10.125.0.2# ..
vrouter running syslog# tls
vrouter running tls#! ca-certificate "-----BEGIN CERTIFICATE-----
... MIID3zCCAkegAwIBAgIIXH6dxQIfVrcwDQYJKoZIhvcNAQELBQAwDTELMAkGA1UE
... AxMCQ0EwHhcNMTkwMzA1MTYwMzE4WhcNMjExMTI4MTYwMzIyWjANMQswCQYDVQQD
... EwJDQTCCAaIwDQYJKoZIhvcNAQEBBQADggGPADCCAYoCggGBAJmOTcw0mfZHwZQG
... K0QM8d0d38x5ABO45sxgiwx5SRwg0jC32Zpqc+b+JyNMH14IUHMYIoxifLEDMtKv
... 0Lg77ARH37cyuqdIDsMkVXI//mgbHx6Qg8Wry0SkgJPby9jRwutz2G49ZtipmrRu
... zXvRjEHrBbfyqvjZmGc2A0Nc1Bp98lViTMWa3BKG9Ym2OTr/PtJpxvYnb85H89fs
... bFjVzQbyyIDFoTmnoaykBzMRGtxzjW/BUL3IzTvHTjFdHzJh7i8OKKyLyepc573p
... b1uTWJJ8Sg8nS46tAU18G+7Y4pYMYh3gGEN9VuiPFV/vzWA7h5dELGOQe3tzSDSS
... 6XnILvQW1yN2R9LQ9ZO8Xl8pEiJ/pwGfcBvIWPHPJDH8TnH3ZcvVwQNt98YwbmUx
... HGYR+2cfP+S6sTvw2ccvz4uENfKVstYTeVRrRHdTpHK7dzUWEU9UAWPpXOufLV69
... Zr6M3fBrBmDUrVaL864kPoDiCMNhtGDhU+Q3nQSvFH8HBTm3zwIDAQABo0MwQTAP
... BgNVHRMBAf8EBTADAQH/MA8GA1UdDwEB/wQFAwMHBAAwHQYDVR0OBBYEFF6FET3m
... 9NiPfbYqwF60m3yGTwXXMA0GCSqGSIb3DQEBCwUAA4IBgQAHWozkh382EAI7i0wW
... CG94WJbxTTNnwA2e6FWqOhSItr4RnfzeHm/DbmfNYlRKYAqGIsjzGLmWz+NozqOg
... Q6qK+RvGjr70zAXuygtRRzi32xuWbAijyx03VRv/FH91F8gf3plR3cNiAhVAW+ef
... xHiGzTrZh8E1HrrIJRj+uoQx66zxkIMZ8nEckxoqs0jFxKY4/7sQ9mQAonlSQg9b
... Y+gJUecbT5Ff2SSyiUCM6XN0FuU/rXglrblscPdFUZeyX24TxWI36qtqYqmJff+d
... aniGfJlJ49Sg3iIoia3zXq7LTl4ZEfSgRhb6V6eDzwX1vhx00O5pGQRwAzwOwBaT
... k+IevZtdlKrEhycvrEoxSH70lPfgHBVJ5QrP8OKkBR5WVaOfcQL/n+703ARepPH9
... NjOAv+HazTPzVDIZwQg+fjftVZtR4CRlaeHGXZy8Tj3SkgiX5SOkTrb0nSHSwoTA
... taiY8eM9lD3siHbuGl/JT3wC8FXvq/cMhougM8NgzgEQ0Zc=
... -----END CERTIFICATE-----"
vrouter running tls#! certificate "-----BEGIN CERTIFICATE-----
... MIIDoTCCAgmgAwIBAgIIXH6fERmi2VkwDQYJKoZIhvcNAQELBQAwDTELMAkGA1UE
... AxMCQ0EwHhcNMTkwMzA1MTYwODUxWhcNMjExMTI4MTYwODUzWjARMQ8wDQYDVQQD
... EwZjbGllbnQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC465HIizN9
... 3qzopW7tD57FjkccmmySPVHWm3dK1dZytSl1ChCPf9rGAXcqagnXFGsjIoKULCWV
... 6eUMa6VQTw4XXIR1dn+x3pfGEp9of/6DR1dzK4UCpXrx4UIKQtDQ1R6UQ8QV1BaI
... ZNvR5XllHD/sS8LUcw8xGilKi+M0x6aPSJAtxoXgX2gn0w3Qn/SzxbCztNizPZO4
... Bk+YEVs6/vK2uyy87tISIkud2HhbxUkckySLawDZxbHr0xTgwcf4jz05GDXMokGx
... dRGVRNeRHfnDS9PZQoyvFeHKmsIrf8VqcMf6qtPGpjBnCm6WDq/rjosD1ZahQ1a7
... MXsoOxg314MJAgMBAAGjgYAwfjAMBgNVHRMBAf8EAjAAMB0GA1UdJQQWMBQGCCsG
... AQUFBwMCBggrBgEFBQcDATAPBgNVHQ8BAf8EBQMDB6AAMB0GA1UdDgQWBBRrHgQ2
... 0vRIncZcD9MKUa6CfUSghjAfBgNVHSMEGDAWgBRehRE95vTYj322KsBetJt8hk8F
... 1zANBgkqhkiG9w0BAQsFAAOCAYEABiDq/MS/YdXiNDE4lcE5A1qy3+S9WjPsx0ql
... zQjr0cN/v/KvEg9Sl8dddtg1HTtdl/Wx57JjrmWymecM5E/HSU/3sxWNHSgjpJsn
... gG4621jwrDWFvuzJHh1jJQyvNa6q++KmI1/UbD9vL+g07Ity2zsRY4vxw6nmDfr5
... Vw6Ml8zh8wD605lJm1AdR20518QfDNhm1mRdAuBacBO00J9/fC0zouOxgSy1W/ha
... 2PUJNf4nxNkQBngfMHKzf/fTmzticQ54Js/LoTkOCEBbhqpajJ//eE6Bx5CIauJN
... dUO6vfryih4wZ5rqsVk57i6lU1jBikvNnrai68MRzst4NUJBi0GVACfQv9efnyEM
... LO3XAMMJUgZNvsrQbVZT1vJnfFehlrxgdXP8c9jiXSFnjZ7SjptxxQzOodaE+2jN
```

(continues on next page)

(continued from previous page)

```
... 7KCJsizW4miGQQyyeBoiQlIF9kjLT5kF41acACTzuPlDxOLo0OG3CB4APDvc8xWo
... J+z2SxQrkTVB5AHT7DyC+Zfjmnu6
... -----END CERTIFICATE-----"
vrouter running tls#! private-key "-----BEGIN RSA PRIVATE KEY-----
... MIIEpAIBAAKCAQEAuOuRyIszfd6s6KVu7Q+exY5HHJpskj1R1pt3StXWcrUpdQoQ
... j3/axgF3KmoJ1xRrIyKClCwllenlDGulUE8OF1yEdXZ/sd6XxhKfaH/+g0dXcyuF
... AqV68eFCCkLQ0NUelEPEFdQWiGTb0eV5ZRw/7EvC1HMPMRopSovjNMemj0iQLcaF
... 4F9oJ9MN0J/0s8Wws7TYsz2TuAZPmBFbOv7ytrssvO7SEiJLndh4W8VJHJMki2sA
... 2cWx69MU4MHH+I89ORg1zKJBsXURlUTXkR35w0vT2UKMrxXhyprCK3/FanDH+qrT
... xqYwZwpulg6v646LA9WWoUNWuzF7KDsYN9eDCQIDAQABAoIBAHWjhw6pX4yHiEBI
... XhT5huvu41ZS9xbhY5q/NFirSM2YalNGn9pqX+bvL7wP0Uq+dpnXbnKM0yxXq5sH
... MBey8yfxd2Kyi/G/xZYAauCz7FnfnMZrvSY918TgpH6amvT/X4C6y5eHYP5MC3uw
... HFYybogIel1lBRkbp4EBFP2StWcYkQd2k7kEAhbk68IKzFLgDf9o6RL8/uSFHVds
... K6946+LRfu0KmMP6QmfI2pGdKwKPiTylVI68SVwQBINLuNOtLPx5zgm1E9wGBghq
... FgBOwHt0vjFOOeql+sjc7MLKv6iR56zDZupnv4rPnz5U9vCv83ApY4BcCmkFPlwd
... A7oPMcECgYEA6OTdIn8A9dG58/jppEdhXOFQSOZjyK4tYMZVdLhg2TU9+0StbePe
... Qf0p8pc/7RtiOwyNu2fpJz3Q5vgfAlUPxSXkxukNfb9uCmAJNmKfYWMtRff/maFU
... E+Vei3MhG7NeR1SfcUEi0zUwW0hhpoIEdrRtWnD5fpWkzdkPh2bT8c0CgYEAy0Q/
... W2A9o3cW8qZHlA3nNT8hoT06v6hKcVVfPKwyG8ql4VYCCYjaXHvxY0sXtweV+2yM
... 8v4sdn0GeVcJUFBri8NJYBTnuRtzTiZfbMDsR7QPwiIf7hyRaQF1KTBUO+givGOp
... XRUa97FUNHkyZUKwyWw8neG+tqOMx28ULz20Ci0CgYEA2bdqCp+T9DmVjr/5Gzwn
... hr6TYTMPwUEi5r9CkBT1ZNjjEoyHXJ2S3zmeB0zh0/Svhegcbz+atLaTHfiCdJm0
... XmcoUdL4a7+TTVuuAQt9V3txjWFjrukkQl1AXzjHkK/DyQcQ7r0noy6sAAnQT+pn
... 5diSCeRnOLEIGe97FudH51kCgYBBhGn3hfnYKpaW98myixivLP4l/pplFFWKWj4s
... TESKeLMnApX9hML9dGXF33pxYFyTgdWcrRifyITBr7As1v8TOYr5EUPvgk2ULwIr
... B7QhGITLyjwIf+TOt82PzSgZdyVbG7SHcDoVBG9jynzX7rsU8XJIYW8bZ3QFBGS5
... JWZWsQKBgQDDan2dN5URjua5zfJBBt4q92bMbkHlZIzQYpRsPv4vOXzVn6Y13Sy8
... uFmYrar4AOHQhpabIvH6MNpgJHK04g1ZALQrq3JIO+wpq6Mf4wyOXTANvLZCHjm/
... LhhDVcUs1nlM6zofsghgiYAcXQmZdDOrsv7POOg54eOY0/8d2yRCJQ==
... -----END RSA PRIVATE KEY-----"
vrouter running tls#! server-authentication name server.example.org server-backup.
↪example.org
vrouter running tls#
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter> show config xml absolute vrf main logging syslog
<config xmlns="urn:6wind:vrouter">
  <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>
```

(continues on next page)

(continued from previous page)

```
            <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
AxMCQ0EwHhcNMTkwMzA1MTYwMzE4WhcNMjExMTI4MTYwMzIyWjANMQswCQYDVQQD
EwJDQTCCAaIwDQYJKoZIhvcNAQEBBQADggGPADCCAYoCggGBAJmOTcw0mfZHwZQG
K0QM8d0d38x5ABO45sxgiwx5SRwg0jC32Zpqc+b+JyNMH14IUHMYIoxifLEDMtKv
0Lg77ARH37cyuqdIDsMkVXI//mgbHx6Qg8Wry0SkgJPby9jRwutz2G49ZtipmrRu
zXvRjEHrBbfyqvjZmGc2A0Nc1Bp98lViTMWa3BKG9Ym2OTr/PtJpxvYnb85H89fs
bFjVzQbyyIDFoTmnoaykBzMRGtxzjW/BUL3IzTvHTjFdHzJh7i8OKKyLyepc573p
b1uTWJJ8Sg8nS46tAU18G+7Y4pYMYh3gGEN9VuiPFV/vzWA7h5dELGOQe3tzSDSS
6XnILvQW1yN2R9LQ9ZO8Xl8pEiJ/pwGfcBvIWPHPJDH8TnH3ZcvVwQNt98YwbmUx
HGYR+2cfP+S6sTvw2ccvz4uENfKVstYTeVRrRHdTpHK7dzUWEU9UAWPpXOufLV69
Zr6M3fBrBmDUrVaL864kPoDiCMNhtGDhU+Q3nQSvFH8HBTm3zwIDAQABo0MwQTAP
BgNVHRMBAf8EBTADAQH/MA8GA1UdDwEB/wQFAwMHBAAwHQYDVR0OBBYEFF6FET3m
9NiPfbYqwF60m3yGTwXXMA0GCSqGSIb3DQEBCwUAA4IBgQAHWozkh382EAI7i0wW
CG94WJbxTTNnwA2e6FWqOhSItr4RnfzeHm/DbmfNYlRKYAqGIsjzGLmWz+NozqOg
Q6qK+RvGjr70zAXuygtRRzi32xuWbAijyx03VRv/FH91F8gf3plR3cNiAhVAW+ef
xHiGzTrZh8E1HrrIJRj+uoQx66zxkIMZ8nEckxoqs0jFxKY4/7sQ9mQAonlSQg9b
Y+gJUecbT5Ff2SSyiUCM6XN0FuU/rXglrblscPdFUZeyX24TxWI36qtqYqmJff+d
```

(continues on next page)

(continued from previous page)

```
aniGfJlJ49Sg3iIoia3zXq7LTl4ZEfSgRhb6V6eDzwX1vhx00O5pGQRwAzwOwBaT
k+IevZtdlKrEhycvrEoxSH70lPfgHBVJ5QrP8OKkBR5WVaOfcQL/n+703ARepPH9
NjOAv+HazTPzVDIZwQg+fjftVZtR4CRlaeHGXZy8Tj3SkgiX5SOkTrb0nSHSwoTA
taiY8eM9lD3siHbuGl/JT3wC8FXvq/cMhougM8NgzgEQ0Zc=
-----END CERTIFICATE-----</ca-certificate>
          <certificate>-----BEGIN CERTIFICATE-----
MIIDoTCCAgmgAwIBAgIIXH6fERmi2VkwDQYJKoZIhvcNAQELBQAwDTELMAkGA1UE
AxMCQ0EwHhcNMTkwMzA1MTYwODUxWhcNMjExMTI4MTYwODUzWjARMQ8wDQYDVQQD
EwZjbGllbnQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC465HIizN9
3qzopW7tD57FjkccmmySPVHWm3dK1dZytSl1ChCPf9rGAXcqagnXFGsjIoKULCWV
6eUMa6VQTw4XXIR1dn+x3pfGEp9of/6DR1dzK4UCpXrx4UIKQtDQ1R6UQ8QV1BaI
ZNvR5XllHD/sS8LUcw8xGilKi+M0x6aPSJAtxoXgX2gn0w3Qn/SzxbCztNizPZO4
Bk+YEVs6/vK2uyy87tISIkud2HhbxUkckySLawDZxbHr0xTgwcf4jz05GDXMokGx
dRGVRNeRHfnDS9PZQoyvFeHKmsIrf8VqcMf6qtPGpjBnCm6WDq/rjosD1ZahQ1a7
MXsoOxg314MJAgMBAAGjgYAwfjAMBgNVHRMBAf8EAjAAMB0GA1UdJQQWMBQGCCsG
AQUFBwMCBggrBgEFBQcDATAPBgNVHQ8BAf8EBQMDB6AAMB0GA1UdDgQWBBRrHgQ2
0vRIncZcD9MKUa6CfUSghjAfBgNVHSMEGDAWgBRehRE95vTYj322KsBetJt8hk8F
1zANBgkqhkiG9w0BAQsFAAOCAYEABiDq/MS/YdXiNDE4lcE5A1qy3+S9WjPsx0ql
zQjr0cN/v/KvEg9Sl8dddtg1HTtdl/Wx57JjrmWymecM5E/HSU/3sxWNHSgjpJsn
gG4621jwrDWFvuzJHh1jJQyvNa6q++KmI1/UbD9vL+g07Ity2zsRY4vxw6nmDfr5
Vw6Ml8zh8wD605lJm1AdR20518QfDNhm1mRdAuBacBO00J9/fC0zouOxgSy1W/ha
2PUJNf4nxNkQBngfMHKzf/fTmzticQ54Js/LoTkOCEBbhqpajJ//eE6Bx5CIauJN
dUO6vfryih4wZ5rqsVk57i6lU1jBikvNnrai68MRzst4NUJBi0GVACfQv9efnyEM
LO3XAMMJUgZNvsrQbVZT1vJnfFehlrxgdXP8c9jiXSFnjZ7SjptxxQzOodaE+2jN
7KCJsizW4miGQQyyeBoiQlIF9kjLT5kF41acACTzuPlDxOLo0OG3CB4APDvc8xWo
J+z2SxQrkTVB5AHT7DyC+Zfjmnu6
-----END CERTIFICATE-----</certificate>
          <private-key>-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAuOuRyIszfd6s6KVu7Q+exY5HHJpskj1R1pt3StXWcrUpdQoQ
j3/axgF3KmoJ1xRrIyKClCwllenlDGulUE8OF1yEdXZ/sd6XxhKfaH/+g0dXcyuF
AqV68eFCCkLQ0NUelEPEFdQWiGTb0eV5ZRw/7EvC1HMPMRopSovjNMemj0iQLcaF
4F9oJ9MN0J/0s8Wws7TYsz2TuAZPmBFbOv7ytrssvO7SEiJLndh4W8VJHJMki2sA
2cWx69MU4MHH+I89ORg1zKJBsXURlUTXkR35w0vT2UKMrxXhyprCK3/FanDH+qrT
xqYwZwpulg6v646LA9WWoUNWuzF7KDsYN9eDCQIDAQABAoIBAHWjhw6pX4yHiEBI
XhT5huvu41ZS9xbhY5q/NFirSM2YalNGn9pqX+bvL7wP0Uq+dpnXbnKM0yxXq5sH
MBey8yfxd2Kyi/G/xZYAauCz7FnfnMZrvSY918TgpH6amvT/X4C6y5eHYP5MC3uw
HFYybogIel1lBRkbp4EBFP2StWcYkQd2k7kEAhbk68IKzFLgDf9o6RL8/uSFHVds
K6946+LRfu0KmMP6QmfI2pGdKwKPiTylVI68SVwQBINLuNOtLPx5zgm1E9wGBghq
FgBOwHt0vjFOOeql+sjc7MLKv6iR56zDZupnv4rPnz5U9vCv83ApY4BcCmkFPlwd
A7oPMcECgYEA6OTdIn8A9dG58/jppEdhXOFQSOZjyK4tYMZVdLhg2TU9+0StbePe
Qf0p8pc/7RtiOwyNu2fpJz3Q5vgfAlUPxSXkxukNfb9uCmAJNmKfYWMtRff/maFU
E+Vei3MhG7NeR1SfcUEi0zUwW0hhpoIEdrRtWnD5fpWkzdkPh2bT8c0CgYEAy0Q/
W2A9o3cW8qZHlA3nNT8hoT06v6hKcVVfPKwyG8ql4VYCCYjaXHvxY0sXtweV+2yM
8v4sdn0GeVcJUFBri8NJYBTnuRtzTiZfbMDsR7QPwiIf7hyRaQF1KTBUO+givGOp
XRUa97FUNHkyZUKwyWw8neG+tqOMx28ULz20Ci0CgYEA2bdqCp+T9DmVjr/5Gzwn
hr6TYTMPwUEi5r9CkBT1ZNjjEoyHXJ2S3zmeB0zh0/SvhegcbZ+atLaTHfiCdJm0
XmcoUdL4a7+TTVuuAQt9V3txjWFjrukkQl1AXzjHkK/DyQcQ7r0noy6sAAnQT+pn
5diSCeRnOLEIGe97FudH51kCgYBBhGn3hfnYKpaW98myixivLP4l/pplFFWKWj4s
TESKeLMnApX9hML9dGXF33pxYFyTgdWcrRifyITBr7As1v8TOYr5EUPvgk2ULwIr
```

(continues on next page)

(continued from previous page)

```
B7QhGITLyjwIf+TOt82PzSgZdyVbG7SHcDoVBG9jynzX7rsU8XJIYW8bZ3QFBGS5
JWZWsQKBgQDDan2dN5URjua5zfJBBt4q92bMbkHlZIzQYpRsPv4vOXzVn6Y13Sy8
uFmYrar4AOHQhpabIvH6MNpgJHK04g1ZALQrq3JIO+wpq6Mf4wyOXTANvLZCHjm/
LhhDVcUs1nlM6zofsghgiYAcXQmZdDOrsv7POOg54eOY0/8d2yRCJQ==
-----END RSA PRIVATE KEY-----</private-key>
        </tls>
      </syslog>
    </logging>
  </vrf>
</config>
```

3.1.5 Network interfaces

Interface types

Overview

Turbo IPsec 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-ports` 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"
    ..
network-port pci-b0s9
    pci-bus-addr 0000:00:09.0
    vendor "Red Hat, Inc"
    model "Virtio network device"
    ..
network-port pci-b0s8
    pci-bus-addr 0000:00:08.0
    vendor "Red Hat, Inc"
    model "Virtio network device"
    ..
vrouter running physical eth0#!
```

Use the `port` command to associate a port identifier to the Ethernet interface:

```
vrouter running physical eth0#! port pci-b0s8
```

After committing the configuration, we can fetch the state of the interface using the following command:

```
vrouter running physical eth0# commit
vrouter running physical eth0# show state / vrf main interface physical eth0
physical eth0
    mtu 1500
    enabled true
    port pci-b0s8
    oper-status UP
    counters
```

(continues on next page)

(continued from previous page)

```
        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
        ..
    ..
vrouter running physical eth0#
```

The same configuration can be applied using the following 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>
        <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 *|fp| context* 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. 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 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
        ..
```

(continues on next page)

(continued from previous page)

```
    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 overriden if `key input` or `key output` is specified:

```
vrouter running gre tunnel1# key both 3
```

The use of input and output keys is independant: 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.

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 IPsec 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 where directly connected. Therefore, 6WIND Turbo IPsec 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
```

(continues on next page)

(continued from previous page)

```
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 afer 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">
  <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>
```

(continues on next page)

(continued from previous page)

```
            <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
vrouter running vlan vlan-blue#! link-interface eth0
vrouter running vlan vlan-blue# commit
```

The VLAN interface is configured provided `eth0` is configured in VRF main.

Let's fetch the state afer committing this configuration:

```
vrouter running vrf main# interface vlan vlan-blue
vrouter running vlan vlan-blue# show state
vlan vlan-blue
    protocol 802.1q
    ethernet
        mac-address de:ad:de:01:02:03
        ..
    mtu 1500
    counters
```

(continues on next page)

(continued from previous page)

```
        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:

```
vrouter> show config xml absolute vrf main interface vlan vlan-blue
<config xmlns="urn:6wind:vrouter">
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <vlan xmlns="urn:6wind:vrouter/vlan">
        <name>vlan-blue</name>
        <protocol>802.1q</protocol>
        <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*:

```
vrouter running vrf main# interface vlan vlan-green
vrouter running vlan vlan-green#! vlan-id 400
vrouter running vlan vlan-green#! link-interface eth0
vrouter running vlan vlan-green# link-vrf vrf1
vrouter running vlan vlan-green# commit
```

VXLAN

Virtual eXtensible Local Area Networks (VLAN) 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':

```
vrouter running vrf main# interface vxlan vxlan100
vrouter running vxlan vxlan100#! vni 100
vrouter running vxlan vxlan100# link-interface eth0
vrouter running vxlan vxlan100# group '239.0.0.8'
vrouter running vxlan vxlan100# commit
```

The VXLAN interface is configured provided `eth0` is configured in VRF main.

Let's fetch the state afer committing this configuration:

```
vrouter running vrf main# interface vxlan vxlan100
vrouter 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
```

(continues on next page)

(continued from previous page)

```
    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:

```
vrouter> 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>
        <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>
```

(continues on next page)

(continued from previous page)

```
        <group>239.0.0.8</group>
      </vxlan>
    </interface>
  </vrf>
</config>
```

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
lag lag0
    mii-link-monitoring 100
    mode lacp
    xmit-hash-policy layer2+3
    mtu 1500
    lacp-rate slow
    oper-status DOWN
    link-interface eth0
```

(continues on next page)

(continued from previous page)

```
    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>
        <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`.

```
vrouter running vrf main# interface bridge br0
vrouter running bridge br0# link-interface eth0
vrouter running bridge br0# link-interface eth1
vrouter 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:

```
vrouter running vrf main# interface bridge br0
vrouter running bridge br0# show state
bridge br0
    oper-status UNKNOWN
    enabled true
    mtu 1500
    link-interface eth0
    link-interface eth1
    ethernet
        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:

```
vrouter 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 loopack 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 (Internet Protocol version 4) 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
```

(continues on next page)

(continued from previous page)

```
    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">
  <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.

SVTI

Secure Virtual Tunnel Interfaces are generic virtual interfaces ensuring IPsec transformation. They are used to configure route-based VPNs.

Each SVTI (Secure Virtual Tunnel Interface) interface has its own SAD (Security Association Database) and SPD (Security Policy Database). These interfaces have an SVTI ID parameter to associate them to IPsec SA/SP. This ID must be unique per-VRF.

To configure SVTI, enter the context `interface` type `svti` from the VRF in which you plan to define the SVTI interface. 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`:

```
vrouter running vrf main# interface svti svti100
vrouter running svti svti100#! svti-id 100
vrouter running svti svti100# commit
```

The SVTI interface is configured and ready to be associated to an IKE VPN.

Let's fetch the state afer committing this configuration:

```
vrouter running vrf main# interface svti svti100
vrouter 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:

```
vrouter> 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>
```

(continues on next page)

(continued from previous page)

```
      </svti>
    </interface>
  </vrf>
</config>
```

Cross-VRF

SVTI interfaces can be used to do cross-VRF; the interface can be located in a VRF and have a different link-VRF where the SA (Security Association) / SP (Security Policy) are located.

Here is an example of an SVTI located in `vrf2` but with a `link-vrf` on `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.

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
```

(continues on next page)

(continued from previous page)

```
        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 (Internet Protocol) 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` afer 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
        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.

```
vrouter 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>
```

(continues on next page)

(continued from previous page)

```
  </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>
        <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.

```
vrouter running config# / vrf vr1
vrouter running vrf vr1# interface xvrf to-vr2
vrouter running xvrf to-vr2#! link-interface to-vr1 link-vrf vr2
vrouter running xvrf to-vr2#! / vrf vr2
vrouter running vrf vr2#! interface xvrf to-vr1
```

(continues on next page)

(continued from previous page)

```
vrouter running xvrf to-vr1#! link-interface to-vr2 link-vrf vr1
vrouter 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:

```
vrouter running config# vrf vr2
vrouter running vrf vr2# routing static
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` afer 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>
```

(continues on next page)

(continued from previous page)

```
    </interface>
  </vrf>
  <vrf>
    <name>vr2</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <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 L3VPN 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
```

(continues on next page)

(continued from previous page)

```
        ..
    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 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>
        <name>eth0</name>
        <mtu>2000</mtu>
        (...)
      </physical>
```

(continues on next page)

(continued from previous page)

```
    </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
```

See also:

The *command reference* for details about the API, and the *show interface* command.

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">
      <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>
```

(continues on next page)

(continued from previous page)

```
          </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 broacast-address
vrouter running dhcp# del request time-offset
```

(continues on next page)

(continued from previous page)

```
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
    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>
```

(continues on next page)

(continued from previous page)

```
            <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>
```

(continues on next page)

(continued from previous page)

```
            <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>
```

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.

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 now that:

- source NAT happens in `postrouting` chain, after `mangle` table
- destination NAT happens in `prerouting` chain, after `raw` and `mangle` tables
- connection tracking 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>
            <value>2.2.2.2</value>
          </address>
        </translate-to>
      </source-rule>
```

(continues on next page)

(continued from previous page)

```
    </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:

```
vrouter running config# vrf main
vrouter running vrf main# nat
vrouter running nat# source-rule 1 outbound-interface public translate-to output-
↪address
vrouter running nat# commit
```

To display the applied configuration:

```
vrouter 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:

```
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>
        <outbound-interface>
          <name>public</name>
        </outbound-interface>
        <translate-to>
          <output-address/>
        </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>
        </protocol>
        <destination>
          <port>
            <value>8080</value>
          </port>
        </destination>
        <inbound-interface>
          <name>public</name>
        </inbound-interface>
        <translate-to>
          <address>
            <value>2.2.2.2</value>
```

(continues on next page)

(continued from previous page)

```
            <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
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
```

(continues on next page)

(continued from previous page)

```
    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

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 IPsec 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# ipv4-route 10.200.0.0/24 next-hop 10.175.0.2 table 100
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

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>
        <ipv4-route>
          <destination>10.200.0.0/24</destination>
          <next-hop>
            <next-hop>10.175.0.2</next-hop>
            <table>100</table>
          </next-hop>
        </ipv4-route>
      </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 (Bidirectional Forwarding Detection) 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 (Internet Protocol version 6) Access List:

```
vrouter running config# routing
vrouter running routing# ipv6-access-list ACCESS-LIST-NAME {permit|deny} X:X::X:X/
↪M [exact-match]
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#
```

ROUTEMAP-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 an other 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:

```
vrouter running routing# route-map ROUTEMAP-NAME seq SEQ policy {permit|deny}
vrouter running route-map SEQ# match peer A.B.C.D
vrouter running route-map SEQ# set weight (0-4294967295)
```

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 RIPng	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 (Internet Service Providers), 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.

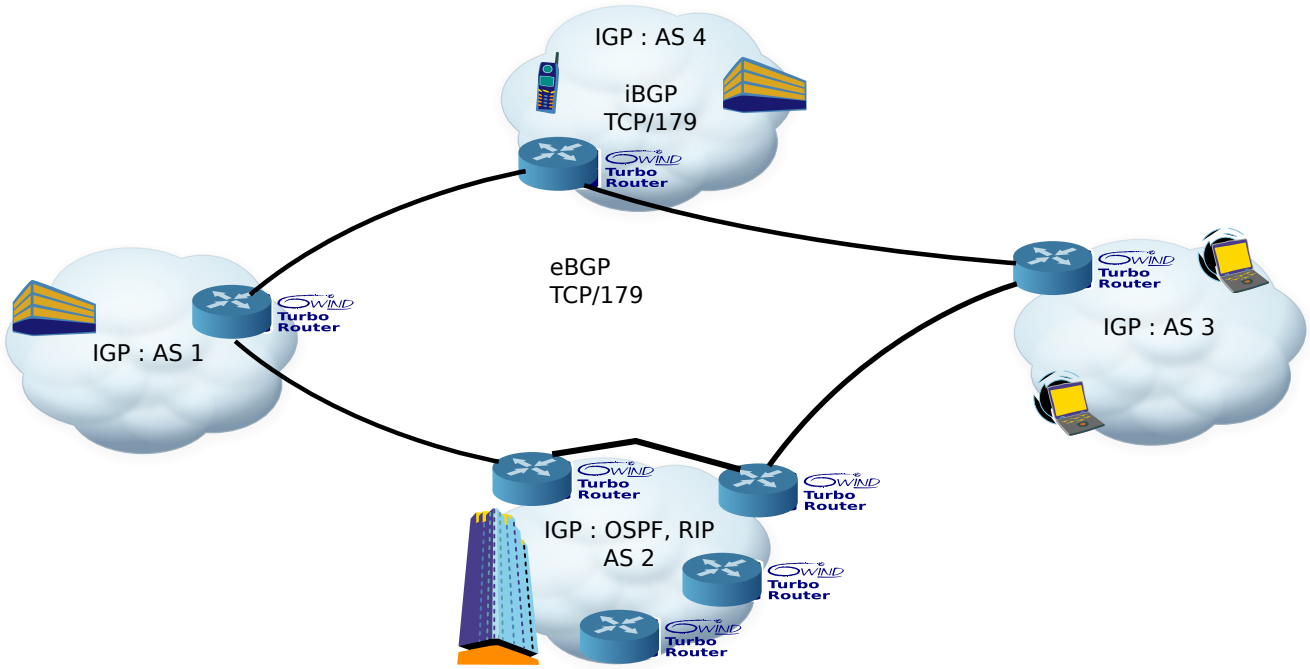


Fig. 1: The EGP (BGP) vs. the IGPs (Internal Gateway Protocols) (RIP, RIPNG, OSPF v2, OSPF v3)

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 IPsec 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

See also:

The *command reference* for details.

BGP configuration

There are a list of necessary elements to know when forging a BGP configuration.

- *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>
```

(continues on next page)

(continued from previous page)

```
            </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
```

(continues on next page)

(continued from previous page)

```
    routing bgp enabled true
    commit
```

Basic BGP configuration

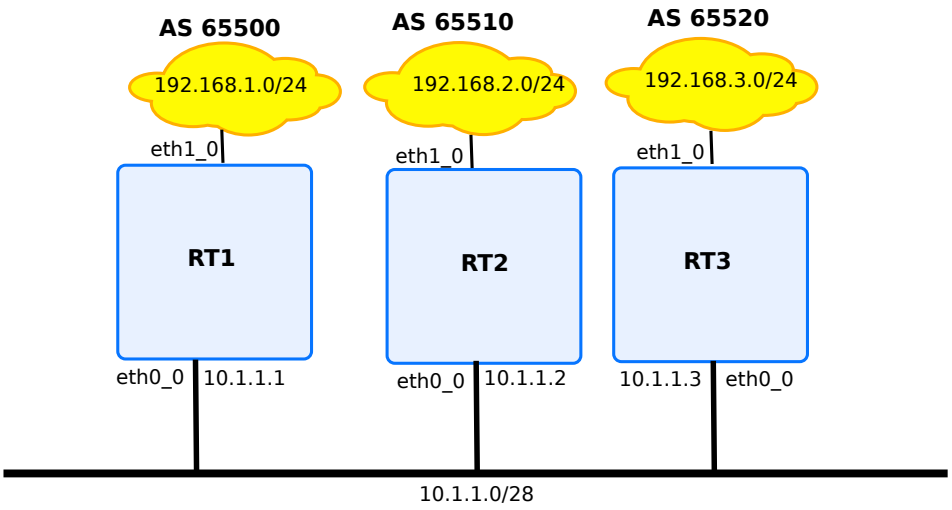


Fig. 2: BGP configuration illustration with 3 BGP peerings

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
```

(continues on next page)

(continued from previous page)

```
       ..
    ..
    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
  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
```

(continues on next page)

(continued from previous page)

```
      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 ?
 *>                  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
```

(continues on next page)

(continued from previous page)

```
         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 (Equal Cost Multi Path) 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
```

(continues on next page)

(continued from previous page)

```
            . .
          . .
        . .
      . .
```

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*
- *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*.

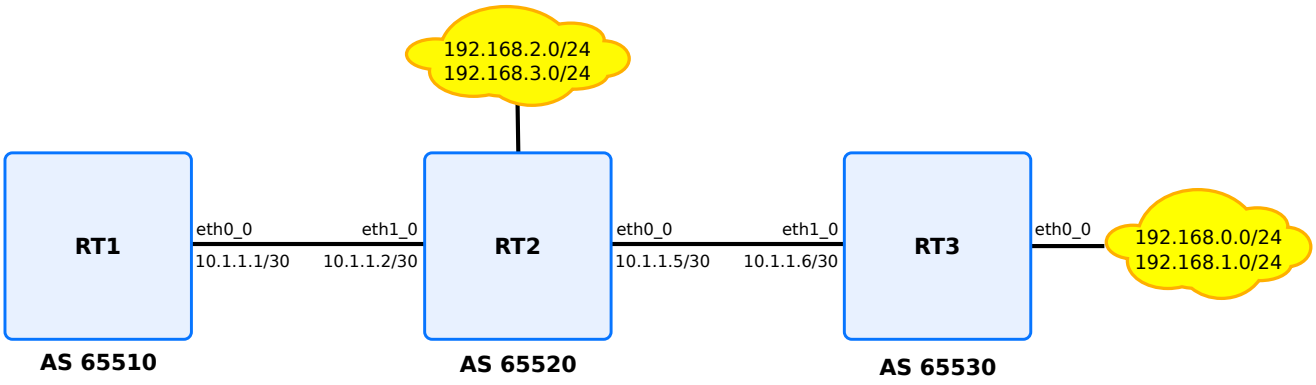


Fig. 3: BGP aggregation

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
           ..
```

(continues on next page)

(continued from previous page)

```
        aggregate-address 192.168.0.0/22
        ..
     ..
  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,
        > - selected route, * - FIB 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
```

(continues on next page)

(continued from previous page)

```
 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 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:

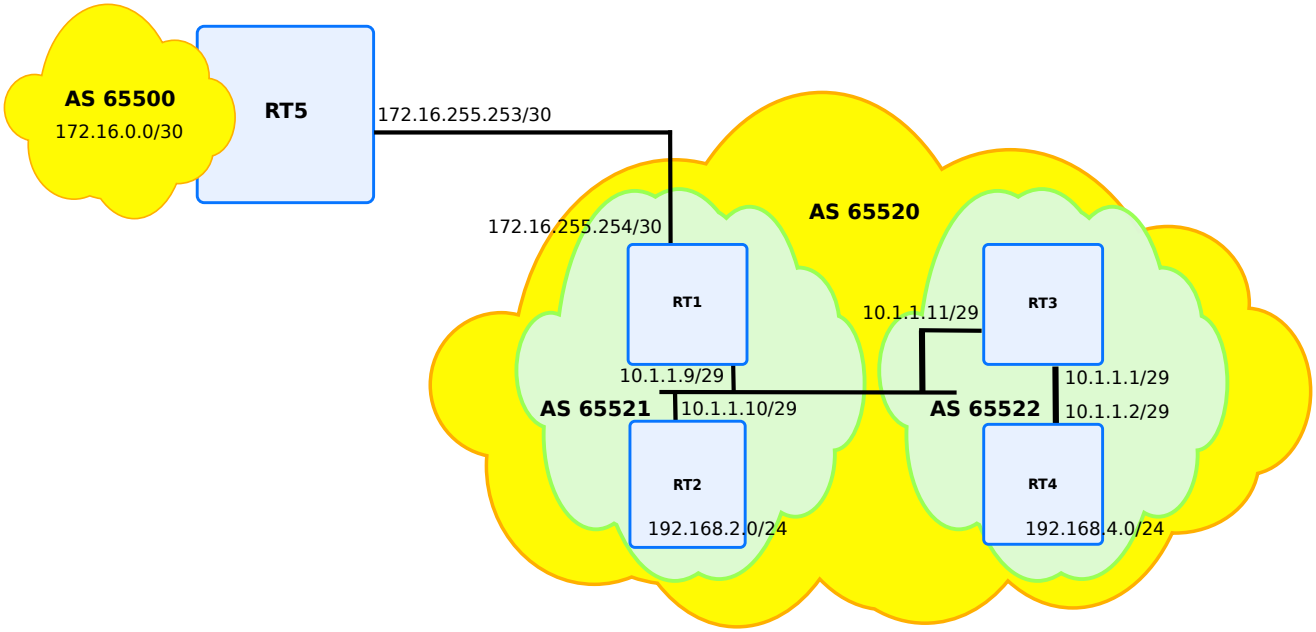


Fig. 4: BGP confederation

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
      ..
```

(continues on next page)

(continued from previous page)

```
   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
 Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
 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,
         > - selected route, * - FIB 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 recursives 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 hiddent 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
      ..
   ..
 ..
```

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 accoringly.

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

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
    .. .. ..
  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 0xFFFFFF01.

no-advertise The routes must not be advertised to any peer. Value is 0xFFFFFF02.

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 0xFFFFFF03.

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 serviecs 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 commnunity):

```
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 extented 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.

The syntax of community list is:

```
routing bgp
  community-list NAME
    policy PRIORITY permit|deny attr1[ attr2[ ...]]
```

NAME The community's name.

PRIORITY The policy priority. Lesser is the value, greater is the priority.

attr1[attr2[...]] Community attribute(s): can take either value under format `AA:BB`, where `AA` and `BB` are (0-65535), or can take one of the following keywords: `local-AS`, `no-advertise`, `no-export`, `internet`.

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
```

(continues on next page)

(continued from previous page)

```
        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.

An extended community list is used in a match clause of a *route map*. An 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.

The syntax of extended community list is:

```
routing bgp
  extcommunity-list NAME
    policy PRIORITY permit|deny soo|rt attr1[ soo|rt attr2[ ...]]
```

NAME The community's name.

PRIORITY The policy priority. Lesser is the value, greater is the priority.

soo|rt attr1[soo|rt attr2[. . .]] Extended community attribute(s) in the format: `soo|rt AS2B:BBBB|AS4B:BB|AS2B:IP`, where `AS2B and `AS4B` respectively stand for the AS 2 byte and AS 4 byte value. `BBBB` and `BB` stand for a 4 and 2 byte value, while `IP` stands for an IPv4 address. `rt` stands for the Route Target extended community.

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
        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). 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 `show-traffic`, 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).
- Layer 4 information for UDP (User Datagram Protocol), TCP : source port, or destination port, or any port.
- Layer 4 information for ICMP type and ICMP code.
- Layer 3 information : DSCP (Differentiated Services Code Point) value, Protocol type, packet length, fragmentation.
- Misc layer 4 TCP flags.

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)

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
```

(continues on next page)

(continued from previous page)

```
    neighbor 1.0.0.1 remote-as 2
    neighbor 1.0.0.1 address-family ipv4 flowspec
    ..
```

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 vrf0
   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 analyser
  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 analyser
    enabled true
    link-interface main
    link-vrf analyser
    ..
  ..
  routing
    bgp
      router-id 192.168.0.162
      as 65100
      neighbor 192.168.0.161
```

(continues on next page)

(continued from previous page)

```
        remote-as 65100
        address-family
          ipv4-flowspec
          ..
        ..
      ..
    ..
  ..
vrf analyser
  interface xvrf main
    link-interface analyser
    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 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 1700000, 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 `analyser` looks like, since a default route in table `257` has been installed to reach `analyser`.

```
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,
             > - selected route, * - FIB route
B>* 0.0.0.0/0 [20/0] is directly connected, analyser, 19:07:48

router> show ipv4-routes vrf analyser
```

(continues on next page)

(continued from previous page)

```
             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 analyser:
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 customer 1
  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
```

(continues on next page)

(continued from previous page)

```
*> 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 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 to carry tunneling information through L3VPN.

L3VPN overview

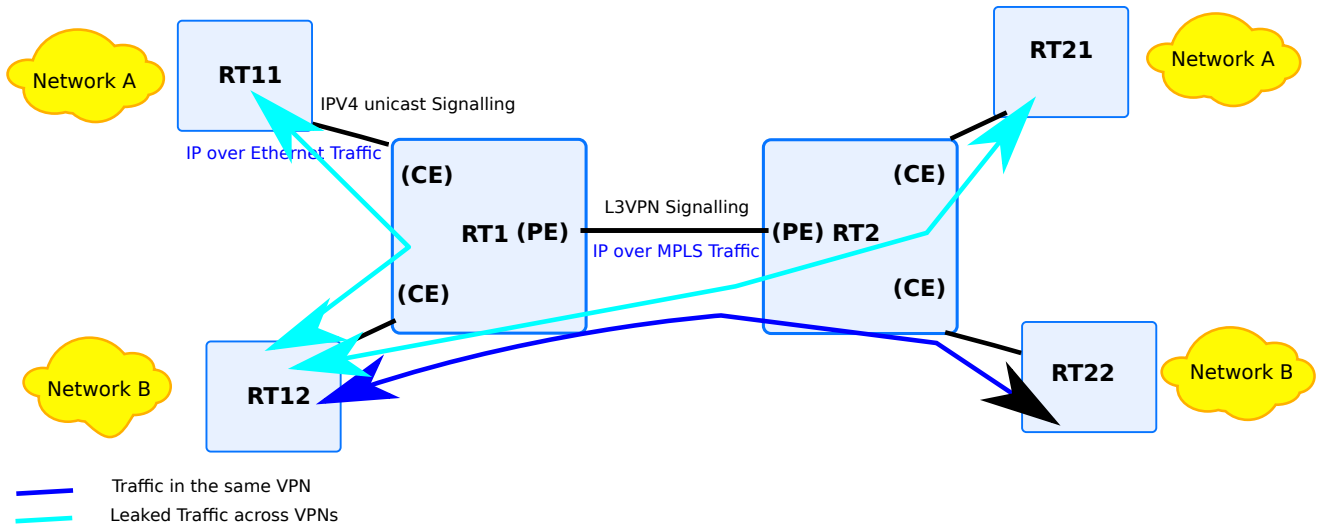


Fig. 5: BGP l3vpn use case example

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 (Virtual Private Networks). 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.
- 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 rout 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.

```
vrf main
  routing bgp
    as 65500
    ..
    ..
```

(continues on next page)

(continued from previous page)

```
  ..
vrf customer1
  routing bgp
    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 11:22
      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.

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. More exactly, BGP L3VPN will append a label that will stand for the encapsulation between 2 VPNs, in one 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 LDP label. More information on chapter (*LDP configuration*).

```
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
```

(continues on next page)

(continued from previous page)

```
   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
[..]
*> 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 negoatiated 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
      ..
      ..
    ..
    ..
  ..
```

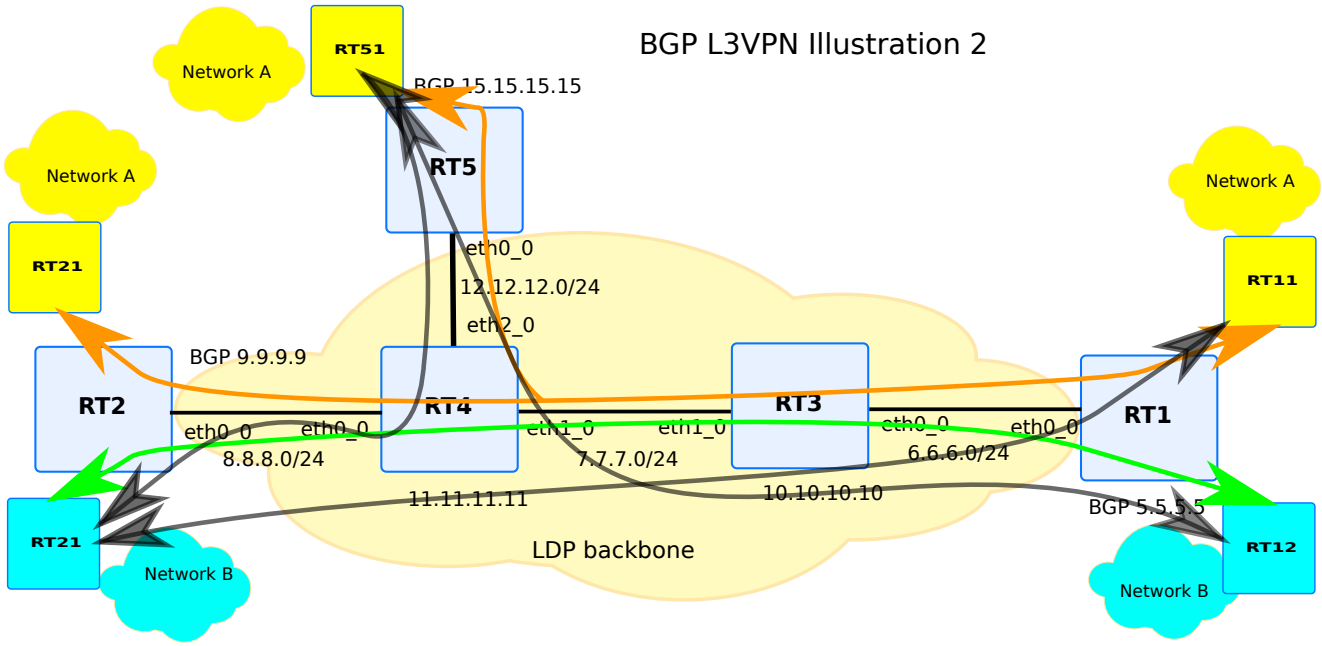



Fig. 6: L3VPN setup using MPLS based framework

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
    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
    ..
    ..
  ..
```

(continues on next page)

(continued from previous page)

```
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
    ..
    ..
  ..
```

(continues on next page)

(continued from previous page)

```
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
      ..
      ..
    ..
    ..
  ..
```

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 32, local router ID is 5.5.5.5, 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
*> 10.101.0.0/24    1.1.1.2@2<              100      0 i
    UN=1.1.1.2 EC{1:55} label=80 type=bgp, subtype=5
*> 10.101.1.0/24    1.1.1.2@2<              100      0 i
    UN=1.1.1.2 EC{1:55} label=80 type=bgp, subtype=5
```

(continues on next page)

(continued from previous page)

```
*> 10.101.2.0/24    1.1.1.2@2<              100      0 i
    UN=1.1.1.2 EC{1:55} label=80 type=bgp, subtype=5
[..]
*>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
[..]
*>i10.101.22.0/24   15.15.15.15              100      0 i
    UN=15.15.15.15 EC{1:55} label=300 type=bgp, subtype=0
*>i10.101.23.0/24   15.15.15.15              100      0 i
    UN=15.15.15.15 EC{1:55} label=300 type=bgp, subtype=0
*>i10.101.24.0/24   15.15.15.15              100      0 i
    UN=15.15.15.15 EC{1:55} label=300 type=bgp, subtype=0
[..]
Route Distinguisher: 2:55
*> 10.201.0.0/24    2.2.2.3@1<              100      0 i
    UN=2.2.2.3 EC{2:55} label=81 type=bgp, subtype=5
*> 10.201.1.0/24    2.2.2.3@1<              100      0 i
    UN=2.2.2.3 EC{2:55} label=81 type=bgp, subtype=5
*> 10.201.2.0/24    2.2.2.3@1<              100      0 i
    UN=2.2.2.3 EC{2:55} label=81 type=bgp, subtype=5
[..]
*>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
[..]
```

As can be seen, all the entries have an underlay nexthop defined (`UN`) that usually stands for the nexthop of the remote BGP global instance. If local entries are imported, that nexthop stands will be sent to remote BGP global instance. Along with the nexthop , an MPLS label will be sent via BGP, and used on top of the MPLS backbone. MPLS stacking will be performed. 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 #PeersVfg #PeersEstb Name      L3-VNI Rmac
DFLT  0      5.5.5.5  2        2          main      0      00:00:00:00:00:00
 VRF  2      1.1.1.1  1        1          customer1 0      00:00:00:00:00:00
 VRF  1      2.2.2.2  1        1          customer2 0      00:00:00:00:00:00
```

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
[..]
*> 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
                           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.

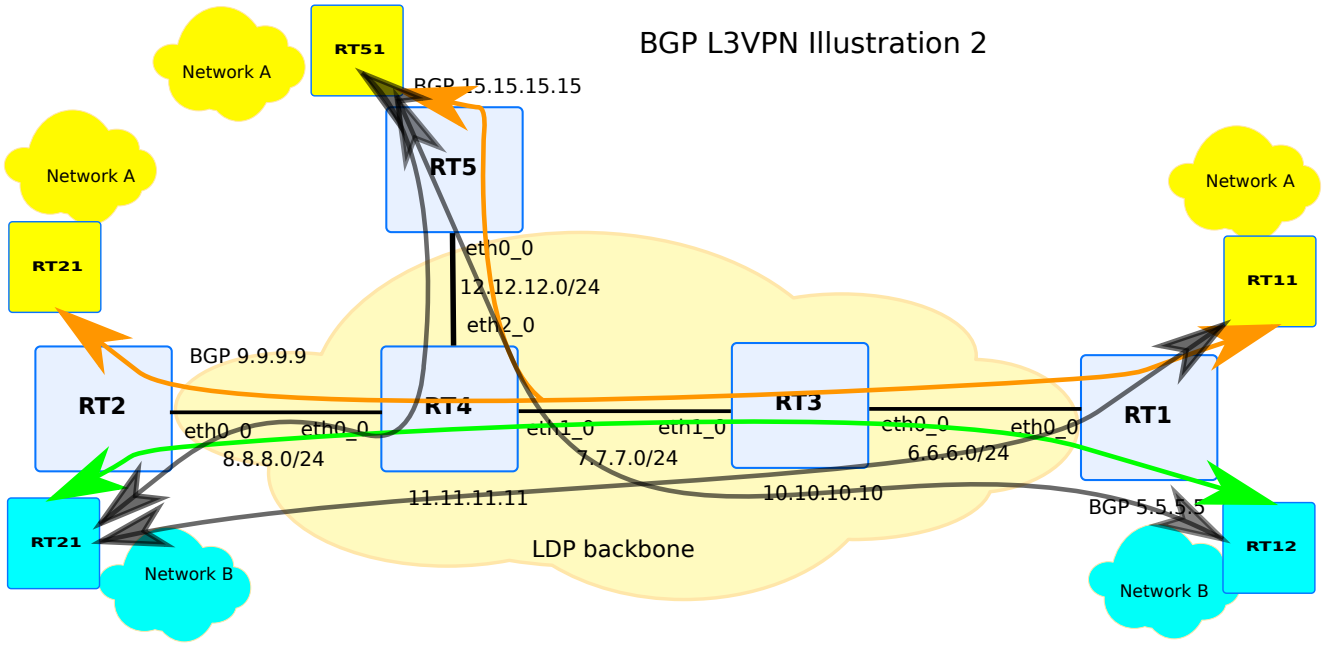


Fig. 7: L3VPN setup using MPLS based framework with VR leaking.

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
    ..
    ..
  ..
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
```

(continues on next page)

(continued from previous page)

```
      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
```

(continues on next page)

(continued from previous page)

```
      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 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
      l3vpn import vpn true
      l3vpn export label 300
```

(continues on next page)

(continued from previous page)

```
      ..
      ..
    ..
    ..
  ..
```

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

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
```

(continues on next page)

(continued from previous page)

```
  *                           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

   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
```

(continues on next page)

(continued from previous page)

```
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
```

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
```

(continues on next page)

(continued from previous page)

```
      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*.

```
vrouter> 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
  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
      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 unrechable, 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
```

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 label information, that is allocated. Then this information is stored in MPLS tables, and it injects that MPLS information in the underlying system so that incoming traffic can benefit from that switching path. 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://frrouting.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 on the connected nodes, then by peering using TCP port 646. Once the TCP session is established, the label information is shared, through label advertisements.

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 a 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 it 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 prevent blackholing during the LDP convergence process, at 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 mean 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 (Forwarding Equivalence Class)) 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

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 `rt3` configurations. The basic neighbour discovery mechanism is used to make the peering work.

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 6.6.6.3
      ipv6-route 10:10::10:10/128 next-hop 6000::3
      ..
    ..
  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
      ..
    ..
```

rt3

```
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 6.6.6.3/24
      ipv6 address 6000::3/64
      ..
    ..
  ..
```

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
Local:
  LSR Id: 5.5.5.5:0
  Transport Address (IPv4): 5.5.5.5
  Transport Address (IPv6): 5:5::5:5
Discovery Sources:
```

(continues on next page)

(continued from previous page)

```
  Interfaces:
    r1-eth2:
      LSR Id: 10.10.10.10:0
          Source address: 6.6.6.3
          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. local bindings are not installed to the underlying system.

```
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
```

(continues on next page)

(continued from previous page)

```
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
```

Among the two remaining entries with the `In Use` Column, only the Remote Label is of interest for local traffic, since this will be the label to be used when forging IP traffic to reach the remote destination. The routing table of the system shows the following:

```
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
[...]
S>* 10.10.10.10/32 [1/0] via 6.6.6.3, 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,
       > - selected route, * - FIB route
[...]
S>* 10:10::10:10/128 [1/0] via 6000::3, eth0_0, label implicit-null, 00:08:19
```

LDP Disabling PHP

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.

```
vrf main
  routing mpls ldp
    router-id 10.10.10.10
    address-family ipv4
      discovery transport-address 5.5.5.5
      label local advertise explicit-null
      interface eth0_0
        ..
      ..
      ..
    ..
    ..
    ..
  routing static
      ipv4-route 11.11.11.11/32 next-hop 10.125.0.2
      ..
```

(continues on next page)

(continued from previous page)

```
    ..
  interface
    loopback loop1
      ipv4 address 10.10.10.10/32
      ..
    physical eth0_0
      port pci-b0s6
      ipv4 address 10.125.0.1/24
      ..
    ..
```

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 10.10.10.10/32       10.10.10.10     16          exp-null         yes
ipv4 10.125.0.0/24        10.10.10.10     exp-null    exp-null          no
ipv4 11.11.11.11/32       10.10.10.10     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 (Bottom Of Stack) bit. In other case will trigger packet drops (as per **RFC 3032** (https://tools.ietf.org/html/rfc3032.html)). 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
      interface eth0_0
        ..
```

(continues on next page)

(continued from previous page)

```
    ..
    ..
  address-family ipv6
    discovery transport-address 10:10::10:10
    interface eth0_0
      ..
    ..
    ..
  ..
```

BackBone LDP configuration

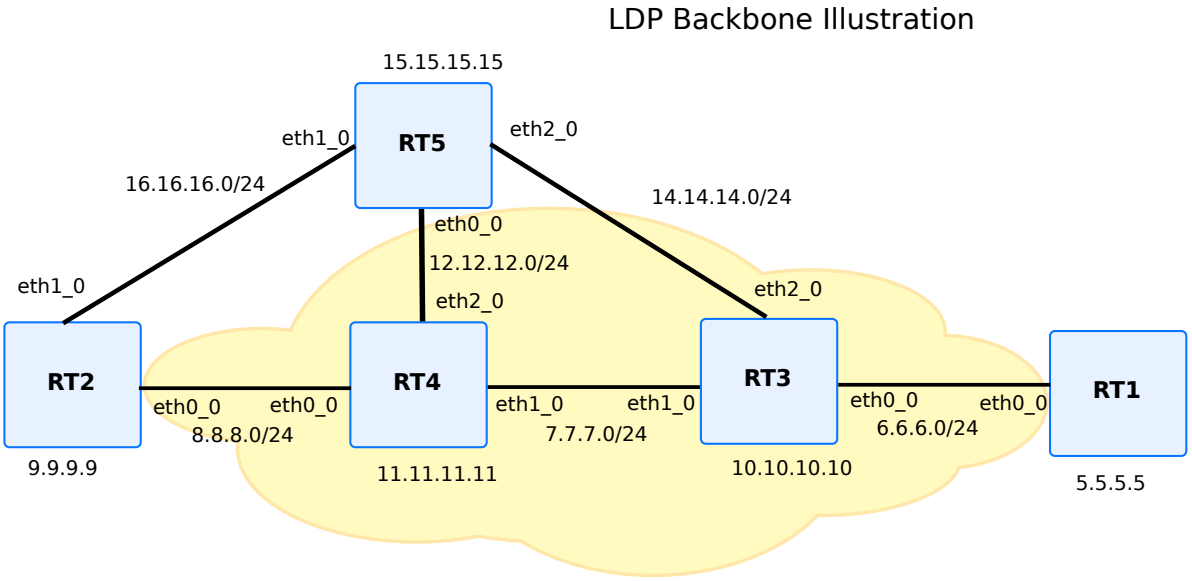


Fig. 8: LDP backbone illustration with multiple nodes, and multiple paths

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
    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
```

(continues on next page)

(continued from previous page)

```
      ..
    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
  address-family ipv4
    discovery transport-address 10.10.10.10
    interface eth0_0
      ..
    interface eth1_0
      ..
    interface eth2_0
      ..
    ..
    ..
  ..
  ..
  ..
interface
```

(continues on next page)

(continued from previous page)

```
  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
  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
```

(continues on next page)

(continued from previous page)

```
    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
  ..
  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
```

(continues on next page)

(continued from previous page)

```
    ..
  ..
  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
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          Nexthop         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,
       > - selected route, * - FIB route

O>* 5.5.5.5/32 [110/30] via 8.8.8.4, r2-eth2, label 21, 00:22:09
```

(continues on next page)

(continued from previous page)

```
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                            Outbound
   Label     Type          Nexthop     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
```

(continues on next page)

(continued from previous page)

```
        ..
      ..
      ..
  ..
  ..
  ..
```

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 everyting 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 if 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

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

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 environnement, 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 Boundary Router) 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

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 supress 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
```

(continues on next page)

(continued from previous page)

```
   This implementation conforms to RFC2328
   RFC1583Compatibility flag is disabled
   OpaqueCapability flag is disabled
   Initial SPF scheduling delay 0 millisec(s)
   Minimum hold time between consecutive SPFs 50 millisec(s)
   Maximum hold time between consecutive SPFs 5000 millisec(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 msecs
   LSA minimum arrival 1000 msecs
   Write Multiplier set to 20
   Refresh timer 10 secs
   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

   ============ 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
```

(continues on next page)

(continued from previous page)

```
    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)

  Link ID         ADV Router      Age  Seq#       CkSum
  10.125.0.3      10.125.0.3      1174 0x80000004 0x57d7
```

OSPF configuration in single area example

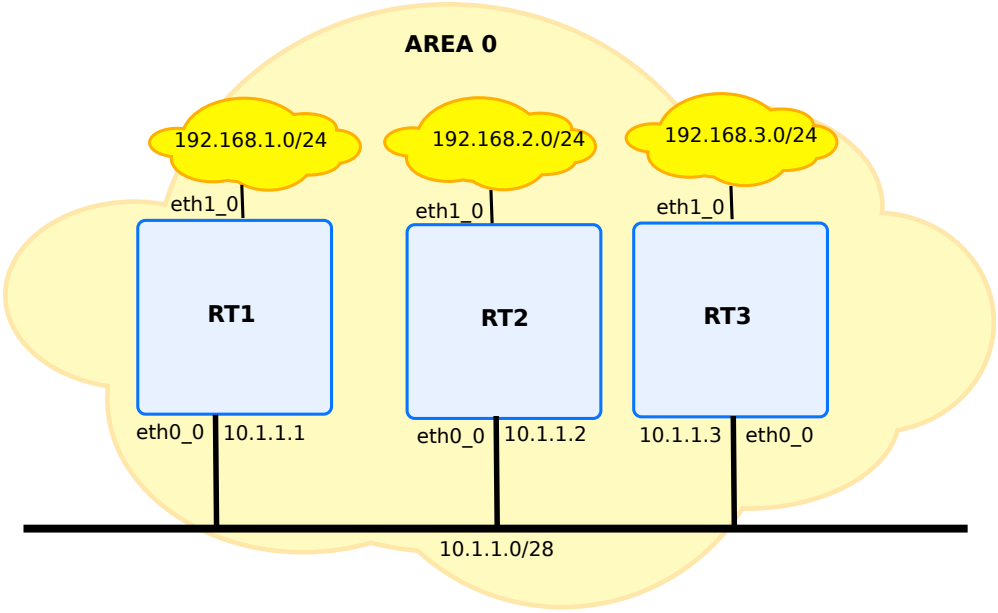


Fig. 9: First OSPF v2 configuration

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
        ..
      ..
    ..
  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
```

(continues on next page)

(continued from previous page)

```
      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
    ..
    ..
```

(continues on next page)

(continued from previous page)

```
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 environnement 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.

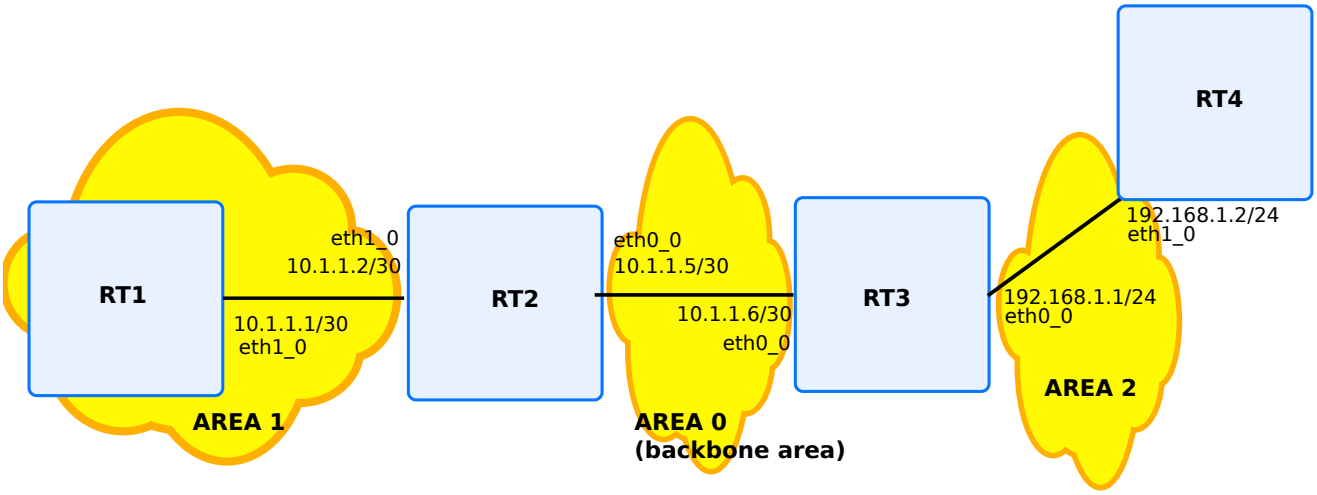


Fig. 10: OSPF v2 router configuration in multi-area environment

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
    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

                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
    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
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 environnement, 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.

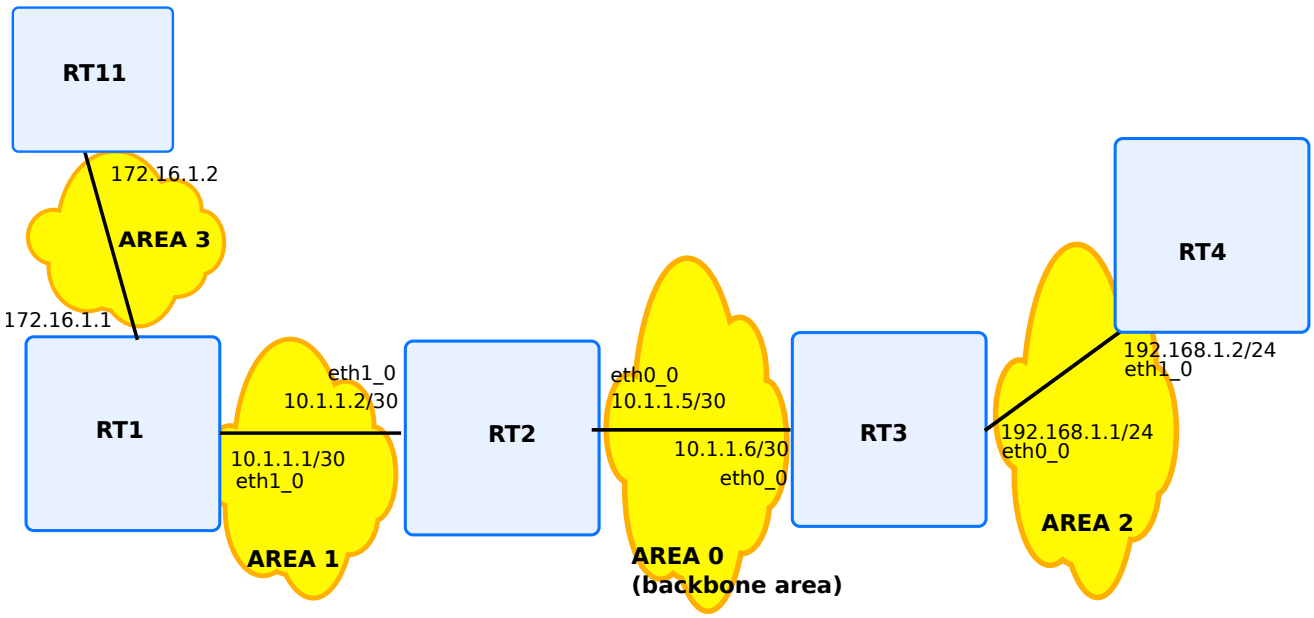


Fig. 11: OSPF v2 virtual link example

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
      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
```

(continues on next page)

(continued from previous page)

```
  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 summmary 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)
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 IPsec 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

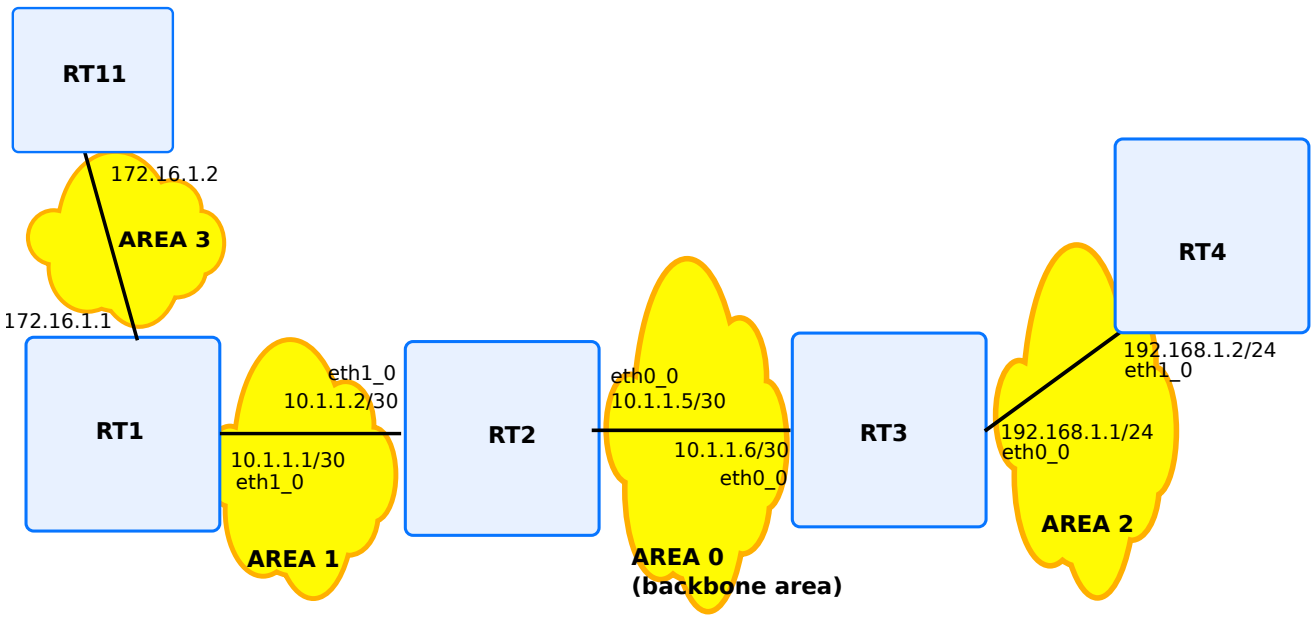


Fig. 12: OSPF v2 options configuration example

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
```

(continues on next page)

(continued from previous page)

```
      ..
    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
   RXmtL RqstL DBsmL
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 0x8000001b  0x827f 2
172.16.1.1      172.16.1.1     598 0x80000010  0x5844 1
192.168.1.1     192.168.1.1    649 0x80000010  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 0x80000001  0x550e

         Summary Link States (Area 0.0.0.0)
```

(continues on next page)

(continued from previous page)

```
Link State      ID ADV Router  Age Seq#         CkSum  Route
10.1.1.0        10.1.1.5       990 0x80000005  0x94a0 10.1.1.0/30
10.1.1.0        172.16.1.1     979 0x80000005  0x7c0b 10.1.1.0/30
172.16.0.0      172.16.1.1     657 0x80000001  0x9b3f 172.16.0.0/22
192.168.1.0     192.168.1.1    626 0x80000006  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 0x80000004  0x35ce 1
172.16.1.1      172.16.1.1     603 0x80000005  0x3e6a 1

          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
```

(continues on next page)

(continued from previous page)

```
                            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 IPsec 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
    ..
    ..
```

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
      ..
```

(continues on next page)

(continued from previous page)

```
    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 0x80000004 0xbf9a 1

            AS External Link States
```

(continues on next page)

(continued from previous page)

```
 Link ID         ADV Router      Age  Seq#       CkSum  Route
 10.100.0.0      1.1.1.1          630 0x80000001 0xc2ff E2 10.100.0.0/24 [0x0]
 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
10.125.0.2      1 Full/Backup       38.091s 10.125.0.2      eth1_0:10.125.0.1
 ↪ 0     0     0
```

(continues on next page)

(continued from previous page)

```
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

   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
```

(continues on next page)

(continued from previous page)

```
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 IPsec. 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 IPsec.

```
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
```

(continues on next page)

(continued from previous page)

```
    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 differents 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 IPsec running state of RIP.

```
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 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 participing 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
```

(continues on next page)

(continued from previous page)

```
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
  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

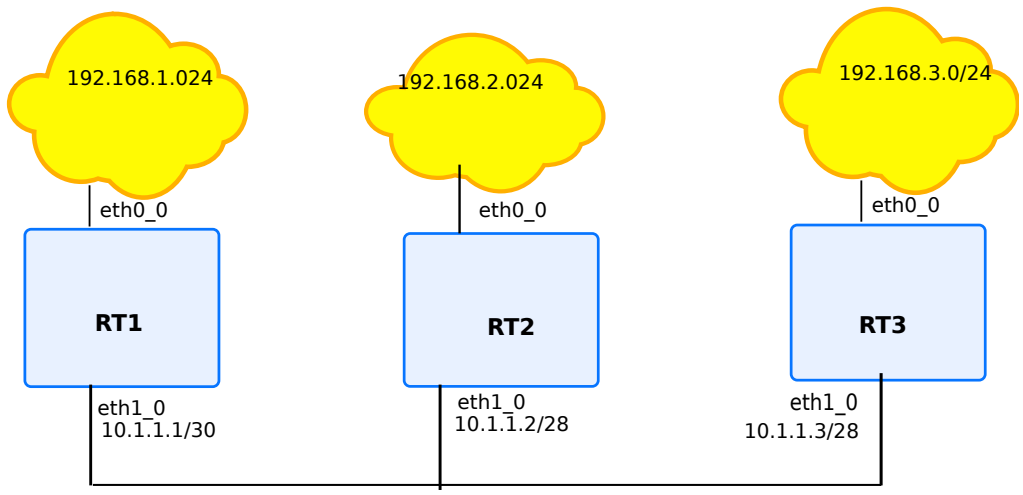


Fig. 13: RIP v2 split-horizon

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 IPsec.

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 IPsec 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.

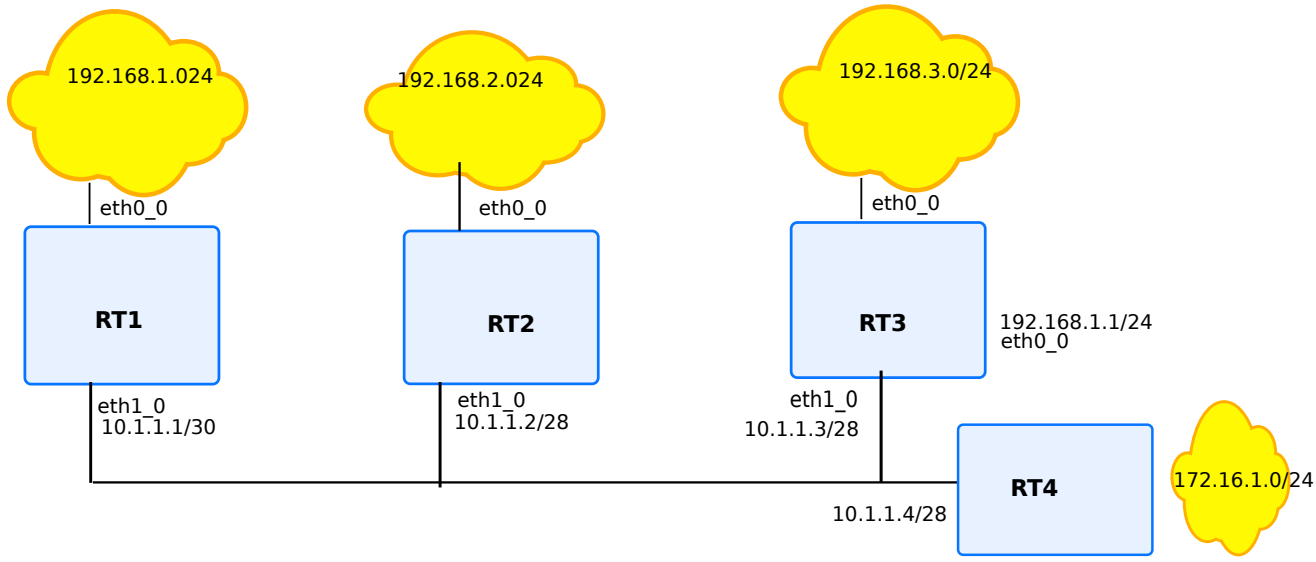


Fig. 14: Next-hop feature

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 (Customer Premise Equipment) 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.

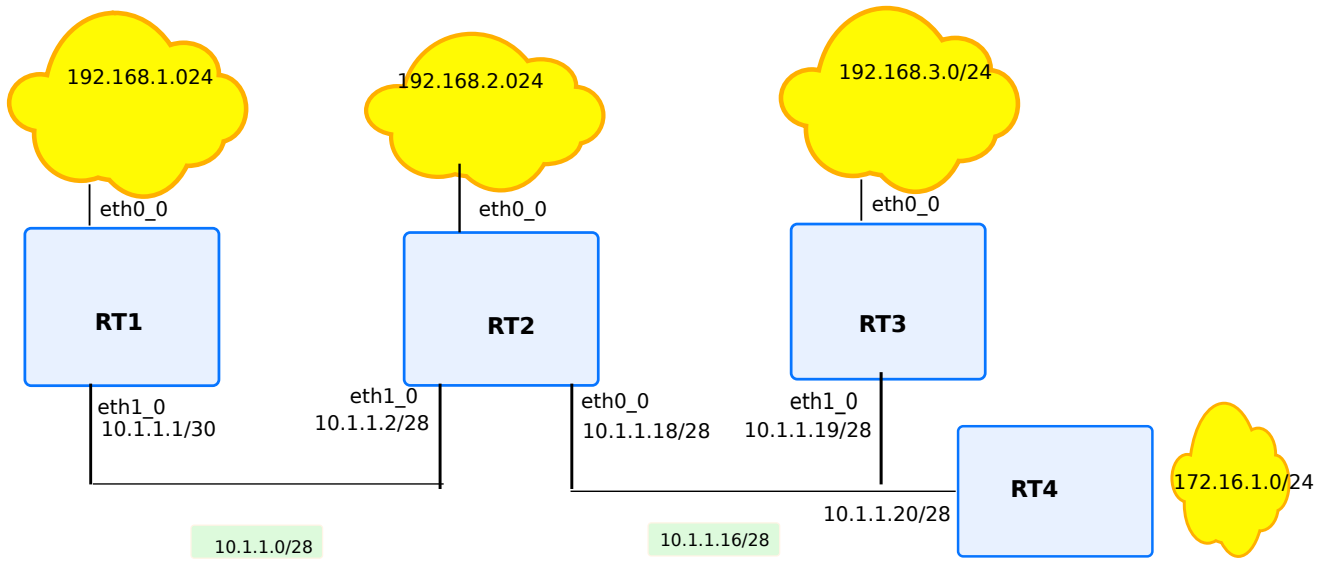


Fig. 15: RIP options

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
      ..
    ..
```

(continues on next page)

(continued from previous page)

```
  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
      ..
    ..
```

rt3

```
vrf main
  interface
    physical eth0_0
      ipv4 address 192.168.3.3/24
      ..
    physical eth1_0
      ipv4 address 10.1.1.19/28
      ..
    ..
```

(continues on next page)

(continued from previous page)

```
  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:
(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 adress 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 *VRRP*. 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 tpye 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

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 msecs
  Initial SPF scheduling delay 0 millisec(s)
  Minimum hold time between consecutive SPFs 50 millsecond(s)
  Maximum hold time between consecutive SPFs 5000 millsecond(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:
```

(continues on next page)

(continued from previous page)

```
 Hello 10, Dead 40, Retransmit 5
DR: 10.1.1.1 BDR: 0.0.0.0
Number of I/F scoped LSAs is 1
  0 Pending LSAs for LSUpdate 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

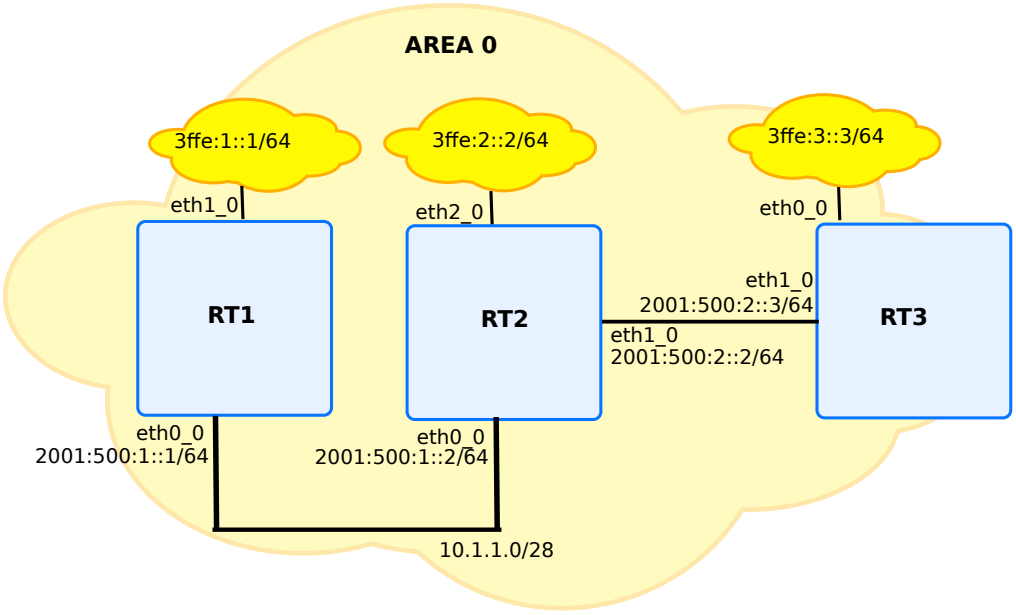


Fig. 16: Basic OSPF v3 configuration

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
```

(continues on next page)

(continued from previous page)

```
      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 neighberhood 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)

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
```

(continues on next page)

(continued from previous page)

```
     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:fef5:88b4

     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
    ..
    ..
  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*.

```
vrouter> 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)
   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


 vrouter> 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]

vrouter> 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
```

(continues on next page)

(continued from previous page)

```
    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, ans 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 IPsec 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:
   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 IPsec.

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
```

(continues on next page)

(continued from previous page)

```
    ..
  ..
..
```

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,
        (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:

```
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
```

(continues on next page)

(continued from previous page)

```
   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:ffff: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.

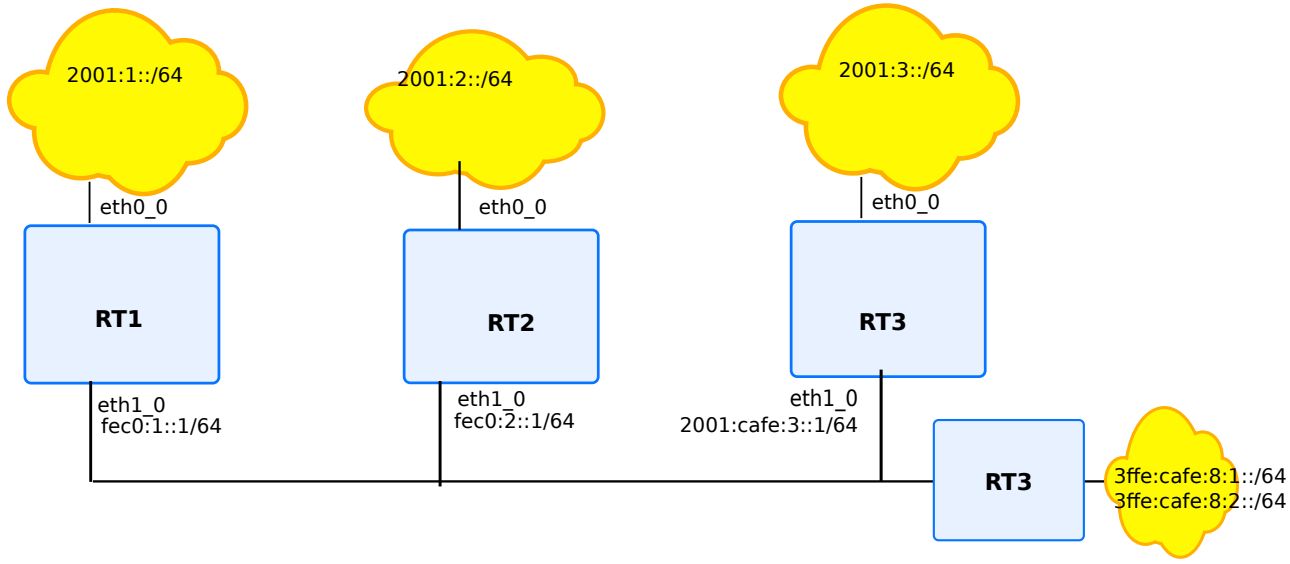


Fig. 17: RIPNG options configuration example

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
      ..
    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 replaceholder 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

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
```

(continues on next page)

(continued from previous page)

```
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.

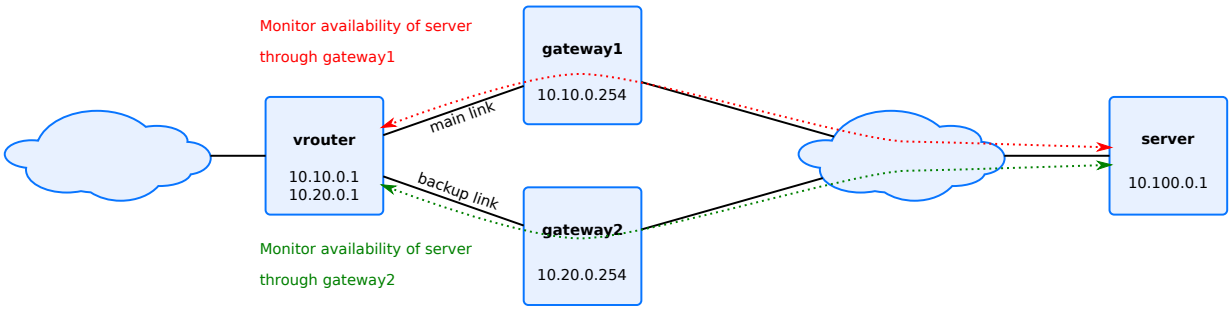
See also:

- Using BFD with BGP, see *Configuring BGP with BFD*.
- Using BFD with OSPF, see *Configuring OSPF with BFD*.
- Using BFD 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. An 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
```

(continues on next page)

(continued from previous page)

```
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
    ..
```

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>
```

(continues on next page)

(continued from previous page)

```
      <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>
          <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/>
      <conntrack/>
    </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>
        <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
```

(continues on next page)

(continued from previous page)

```
vrouter running static# ipv4-route 192.165.1.0/24 next-hop 10.125.0.2 table 100
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
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

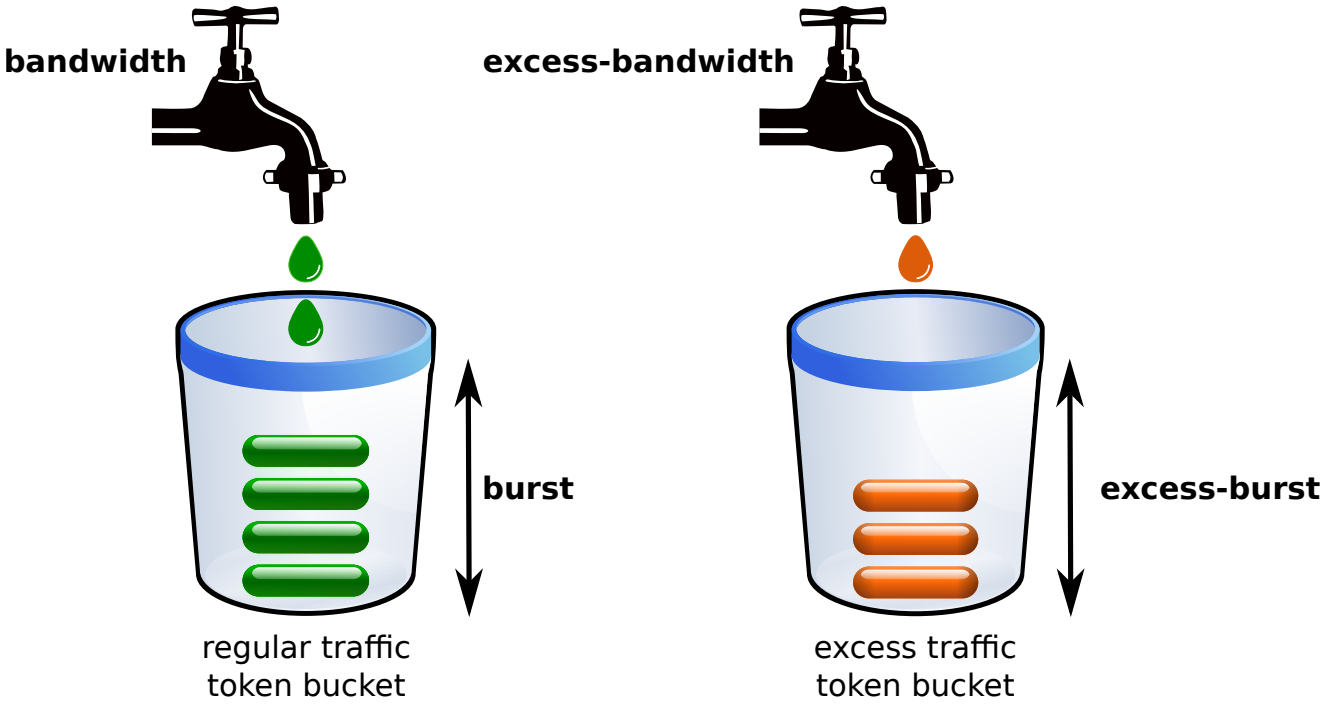
The rate limit of an interface is controlled by a policer, in charge of dropping traffic that does not fulfill a given traffic profile.

The policer specifies the maximum commited 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 (default 1500),
- `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 (default 1500).



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 1500 bytes. Excess traffic is autorized up to 15 Mbps with bursts up to the default 1500 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
        excess-burst 1500
        ..
    policer pol2
        bandwidth 2G
        burst 1500
        excess-bandwidth 15M
        excess-burst 1500
        ..
    ..
```

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>
      <excess-burst>1500</excess-burst>
```

(continues on next page)

(continued from previous page)

```
      <bandwidth>1000000000</bandwidth>
    </policer>
    <policer>
      <name>pol2</name>
      <burst>1500</burst>
      <excess-bandwidth>15000000</excess-bandwidth>
      <excess-burst>1500</excess-burst>
      <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` instanciates 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 1000^2
- `G` (for giga): multiply by 1000^3
- `T` (for tera): multiply by 1000^4

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#
```

(continues on next page)

(continued from previous page)

```
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
        excess-burst 1500
        ..
    shared-policer shared-pol1
        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>
      <excess-burst>1500</excess-burst>
      <bandwidth>1000000000</bandwidth>
    </policer>
    <shared-policer>
      <name>shared-pol1</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 pol1:

```
vrouter running qos# egress rate-limit policer pol1
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 pol1
                ..
            ..
        ..
    ..
```

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
```

(continues on next page)

(continued from previous page)

```
                burst 1500
                excess-bandwidth 0
                excess-burst 1500
                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:

```
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>pol1</policer>
            </rate-limit>
          </egress>
        </qos>
      </physical>
    </interface>
  </vrf>
</config>
```

Each interface that specifies `rate-limit policer pol1` instanciates 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
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
```

(continues on next page)

(continued from previous page)

```
                rate-limit
                    shared-policer shared-pol1
                    ..
                ..
            ..
        ..
```

Commit the configuration:

```
vrouter running interface# commit
Configuration committed.
vrouter running interface# /
vrouter running config#
```

Show interface qos state:

```
vrouter running config# show state vrf main interface
interface
    (...)
    physical eth0
        (...)
        qos
            egress
                rate-limit
                    policer
                        bandwidth 1G
                        burst 2K
                        excess-bandwidth 0
                        excess-burst 1500
                        shared-policer shared-pol1
                        stats
                            pass-packets 0
                            pass-bytes 0
                            pass-excess-packets 0
                            pass-excess-bytes 0
                            drop-packets 0
                            drop-bytes 0
                            ..
                        ..
                    ..
                ..
            ..
        ..
    physical eth1
        (...)
        qos

     qos
         egress
```

(continues on next page)

(continued from previous page)

```
            rate-limit
                policer
                    bandwidth 1G
                    burst 2K
                    excess-bandwidth 0
                    excess-burst 1500
                    shared-policer shared-pol1
                    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>pol1</name>
      <burst>2000</burst>
      <excess-bandwidth>0</excess-bandwidth>
      <excess-burst>1500</excess-burst>
      <bandwidth>1000000000</bandwidth>
    </policer>
    <shared-policer>
      <name>shared-pol1</name>
      <policer>pol1</policer>
    </shared-policer>
  </qos>
  <vrf>
    <name>main</name>
    <interface xmlns="urn:6wind:vrouter/interface">
      <physical>
        <name>eth0</name>
        (...)
        <qos>
          <egress>
            <rate-limit>
              <shared-policer>shared-pol1</shared-policer>
            </rate-limit>
          </egress>
```

(continues on next page)

(continued from previous page)

```
      </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:

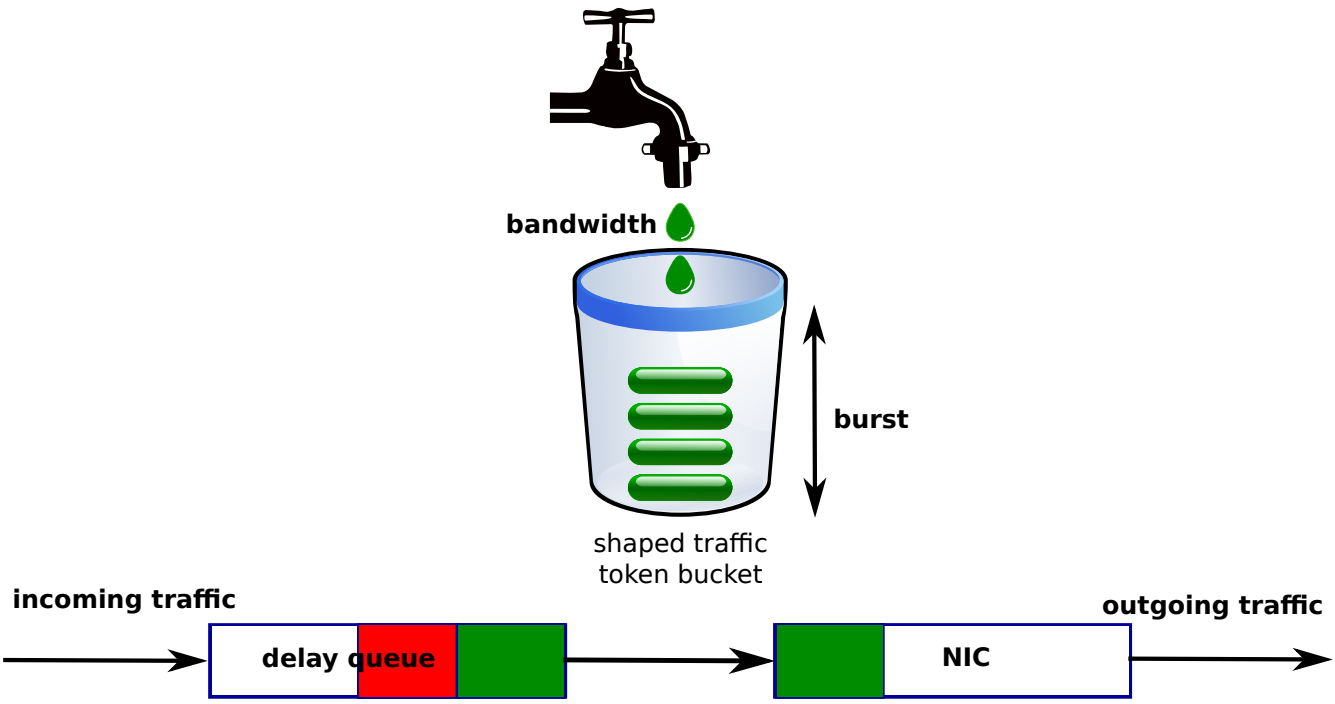
- *bridge interfaces qos*
- *gre interfaces qos*
- *ipip interfaces qos*
- *lag interfaces qos*
- *loopback interfaces qos*
- *physical interfaces qos*
- *veth interfaces qos*
- *vlan interfaces qos*
- *vxlan interfaces qos*
- *xvrf interfaces qos*

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:

```
vrouter running config# qos
vrouter running qos#
vrouter running qos# shaper shaper1
vrouter running shaper shaper1#! bandwidth 1G
vrouter running shaper shaper1# burst 2K
vrouter running shaper shaper1# queue-size 128
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
        burst 48K
        queue-size 256
        layer1-overhead 24
        ..
    ..
```

The same settings can be applied using the following NETCONF XML configuration:

```
vrouter 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>1000000000</bandwidth>
    </shaper>
    <shaper>
      <name>shaper2</name>
      <burst>48000</burst>
      <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:

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

Configure `shaper1` as the rate limiter for egress traffic:

```
vrouter running qos# egress rate-limit shaper shaper1
vrouter running qos# ..
vrouter running physical eth0#
```

Review `eth0` configuration:

```
vrouter running physical eth0# show config nodefault
physical eth0
    (...)
    qos
        egress
            rate-limit
                shaper shaper1
                ..
            ..
        ..
    ..
```

Commit the configuration:

```
vrouter running physical eth0# commit
Configuration committed.
vrouter running physical eth0# /
vrouter running config#
```

Review the QoS state of the interface:

```
vrouter running config# show state vrf main interface physical eth0
physical eth0
    qos
        egress
            rate-limit
                shaper
                    bandwidth 1G
                    burst 2K
                    queue-size 128
                    layer1-overhead 0
                    stats
                        pass-packets 0
                        drop-packets 0
                        ..
                    ..
                ..
            ..
        ..
    ..
..
```

The same settings can be applied 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>
        <qos>
          <egress>
            <rate-limit>
              <shaper>shaper1</shaper>
            </rate-limit>
          </egress>
        </qos>
      </physical>
    </interface>
  </vrf>
```

(continues on next page)

(continued from previous page)

```
</config>
```

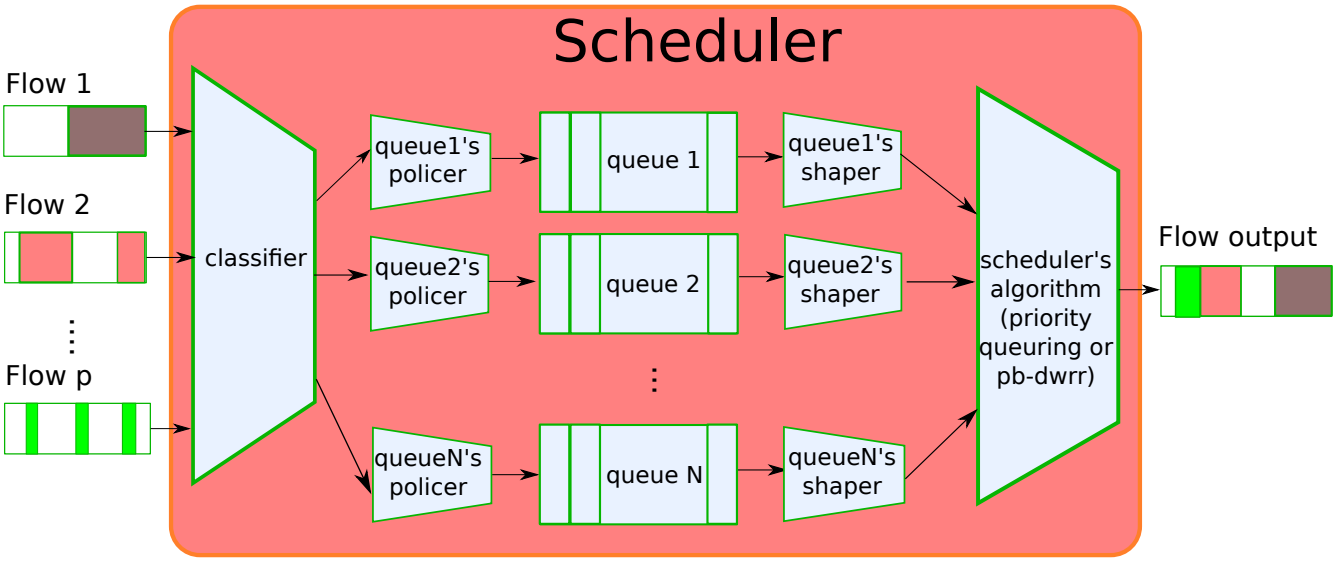
Scheduling

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.
- Specific parameters related to the selected queueing processing (Priority Queueing or PB-DWRR)
- An output shaper to limit the bandwidth used by a queue.
- An input policer to rate limit incoming traffic. Like the output shaper, the purpose of the input policer is to limit the bandwidth used by a queue.
- The list of traffic classes that are submitted to the queue.



Scheduling is applied to egress traffic on physical interfaces.

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.

Traffic classes

A class specifies a set of traffic flows that will be scheduled in the same queue. Classes are defined by the mark of the packet. Packet marking must be done at the *IP packet filtering* level. Refer to the `mark` action of the `rule` command in the *command reference*.

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 (the one with the highest class id).

Classes are created in the global `qos` context with the `class` command. They can then be referenced by any scheduler.

Enter the global `qos` context and create classes:

```
vrouter running config# qos
vrouter running qos#
vrouter running qos# class voip
vrouter running class voip#! mark 0x1
vrouter running class voip# ..
vrouter running qos# class mail
vrouter running class mail#! mark 0x2
vrouter running class mail# ..
vrouter running qos#
```

By default, all bits of the mark are used to specify classes. Therefore, up to 2**32 different classes 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 available classes 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`:

```
vrouter running config# qos
vrouter running qos# class-mask 0xff
vrouter running qos#
```

With this configuration, the first 8 bits of the mark are used to specify classes for QoS, so that 256 classes can be used.

Note: The result of AND operation between class-mask and mark of any specified class must be unique.

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#
vrouter running qos# scheduler sched1
vrouter running scheduler sched1#! pq
vrouter running scheduler pq# nb-queue 3
vrouter running scheduler pq# queue 1
vrouter running scheduler queue 1# class voip
vrouter running scheduler queue 1# shaper shaper1
vrouter running scheduler queue 1# ..
vrouter running scheduler pq# queue 2
vrouter running scheduler queue 2# class mail
vrouter running scheduler queue 2# .. .. ..
vrouter running qos#
```

Enter the global `qos` context and create a scheduler using PB-DWRR:

```
vrouter running config# qos
vrouter running qos#
vrouter running qos# scheduler sched2
vrouter running scheduler sched2#! pb-dwrr
vrouter running scheduler pb-dwrr# nb-queue 3
vrouter running scheduler pb-dwrr# queue 1
vrouter running scheduler queue 1# class voip
vrouter running scheduler queue 1# shaper shaper1
vrouter running scheduler queue 1# priority high
```

(continues on next page)

(continued from previous page)

```
vrouter running scheduler queue 1# ..
vrouter running scheduler pb-dwrr# queue 2
vrouter running scheduler queue 2# class mail
vrouter running scheduler queue 2# quantum 3000
vrouter running scheduler queue 2# .. .. ..
vrouter running qos#
```

Review the QoS configuration:

```
vrouter running qos# show config
qos
    shaper shaper1
        bandwidth 1G
        burst 2K
        layer2-overhead 0
        ..
    scheduler sched1
        pq
            nb-queue 3
            queue 1
                size 256
                shaper shaper1
                class voip
                ..
            queue 2
                size 256
                class mail
                ..
            ..
        ..
    scheduler sched2
        pb-dwrr
            nb-queue 3
            queue 1
                size 256
                shaper shaper1
                class voip
                quantum 1500
                priority high
                ..
            queue 2
                size 256
                class mail
                quantum 1500
                priority low
                ..
            ..
        ..
    class-mask 0xff
```

(continues on next page)

(continued from previous page)

```
    class voip
        mark 0x1
        ..
    class mail
        mark 0x2
        ..
    ..
```

The same settings can be applied using the following NETCONF XML configuration:

```
vrouter running config qos# show config xml absolute
<config xmlns="urn:6wind:vrouter">
  <qos xmlns="urn:6wind:vrouter/qos">
    <class-mask>0xff</class-mask>
    <shaper>
      <name>shaper1</name>
      <burst>2000</burst>
      <layer2-overhead>0</layer2-overhead>
      <bandwidth>1000000000</bandwidth>
    </shaper>
    <class>
      <name>voip</name>
      <mark>0x1</mark>
    </class>
    <scheduler>
      <name>sched1</name>
      <pq>
        <nb-queue>3</nb-queue>
        <queue>
          <id>1</id>
          <class>
            <name>voip</name>
          </class>
          <size>256</size>
          <shaper>shaper1</shaper>
        </queue>
        <queue>
          <id>2</id>
          <class>
            <name>mail</name>
          </class>
          <size>256</size>
        </queue>
      </pq>
    </scheduler>
    <class>
      <name>mail</name>
      <mark>0x2</mark>
    </class>
```

(continues on next page)

(continued from previous page)

```
    <scheduler>
      <name>sched2</name>
      <pb-dwrr>
        <nb-queue>3</nb-queue>
        <queue>
          <id>1</id>
          <class>
            <name>voip</name>
          </class>
          <size>256</size>
          <quantum>1500</quantum>
          <priority>high</priority>
          <shaper>shaper1</shaper>
        </queue>
        <queue>
          <id>2</id>
          <class>
            <name>mail</name>
          </class>
          <size>256</size>
          <quantum>1500</quantum>
          <priority>low</priority>
        </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:

```
vrouter running config# show state vrf main interface physical eth0
qos
    egress
        rate-limit
            shaper
                bandwidth 10G
                burst 48K
                layer2-overhead 24
                stats
                    pass-packets 0
                    drop-packets 0
                    ..
                ..
            ..
        scheduler
            core 1
            pq
                nb-queue 3
                queue 1
                    size 256
                    shaper
                        bandwidth 1G
                        burst 2K
                        stats
```

(continues on next page)

(continued from previous page)

```
                        pass-packets 0
                        drop-packets 0
                        ..
                    ..
                class 0x00000001
                    stats
                        match-packets 0
                        ..
                    ..
                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
                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">
      <physical>
```

(continues on next page)

(continued from previous page)

```
        <name>eth0</name>
        (...)
        <qos>
          <egress>
            <scheduler>sched1</scheduler>
            <rate-limit>
              <shaper>shaper2</shaper>
            </rate-limit>
          </egress>
        </qos>
      </physical>
    </interface>
  </vrf>
</config>
```

3.1.9 Security

IKE

Internet Key Exchange (IKE) is the control plane protocol providing authentication and key exchange mechanisms to establish secure VPNs over IPsec.

Either pre-shared keys or certificates can be used for authentication.

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

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 iketemp1
vrouter running ike-policy-template iketemp1#
```

The IKE policy template is initialized with various default values:

```
vrouter running ike-policy-template iketemp1# show config
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
    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 iketemp1# 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 iketemp1# ..
vrouter running ike#
```

```
vrouter running ike# show config nodefault
ike
    (...)
    ike-policy-template iketemp1
        ike-proposal 1
            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 overridde 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
```

(continues on next page)

(continued from previous page)

```
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
            ..
        ..
    ..
```

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 nodefault
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
        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
            ..
        ..
```

(continues on next page)

(continued from previous page)

```
    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` *iketemp1* and override parameter `keying-tries`, use the `ipsec-policy-template` *ipsectemp1*.

```
vrouter running vpn vpn-hq#! ike-policy
vrouter running ike-policy#! template iketemp1
vrouter running ike-policy#! keying-tries 10
vrouter running ike-policy#! ..
vrouter running vpn vpn-hq#! ipsec-policy
vrouter running ipsec-policy#! template ipsectemp1
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
```

(continues on next page)

(continued from previous page)

```
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 iketemp1
        ike-proposal 1
            enc-alg aes128-cbc
            auth-alg hmac-sha512
            dh-group modp2048
            ..
        ..
    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
            ..
        ..
    ..
```

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 nodefault
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
            ..
        ..
    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 (Extensible Authentication Protocol) 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 overriden in the VPN `ike-policy`:

```
vrouter running ike# vpn vpn-hq
vrouter running vpn vpn-hq# ike-policy
vrouter running ike-policy# local-auth-method certificate
vrouter running ike-policy# remote-auth-method eap-mschapv2
vrouter running ike-policy# ..
vrouter running vpn vpn-hq#
```

If unspecified, the default authentication method is `pre-shared-key`.

The local IKE identity is defined in the VPN:

```
vrouter 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:

```
vrouter 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.

```
vrouter 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:

```
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
ike
    (...)
    pre-shared-key hq-secgw
        id secgw.6wind.net
        secret 0seaJ31RfzHNRvUSH0oUYg7znTW0I=
        ..
```

Each pre-shared key has a name and is composed of two parts, a key and optional IKE identifier selectors (a list of IKE identifiers).

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 IKE identifier selectors `id`, specify 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/IpsecSec selectors).

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 overriden in the VPN `ike-policy`.


```
vrouter running ike# ike-policy-template ikepsk local-auth-method pre-shared-key
vrouter running ike# vpn vpn-hq ike-policy template ikepsk
```

or:

```
vrouter running ike# vpn vpn-hq ike-policy local-auth-method pre-shared-key
```

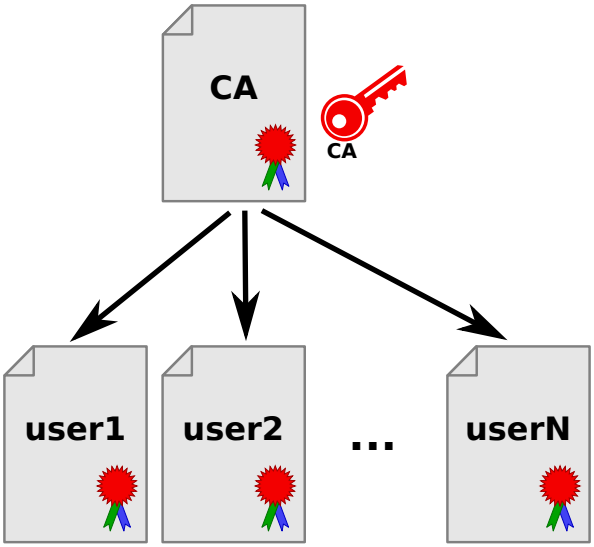
Similarly, to authenticate the remote peer by pre-shared keys, the `remote-auth-method` must be set to `pre-shared-key` in the `ike-policy-template` used by the VPN, or overriden in the VPN `ike-policy`.

Pre-shared keys is the default authentication method.

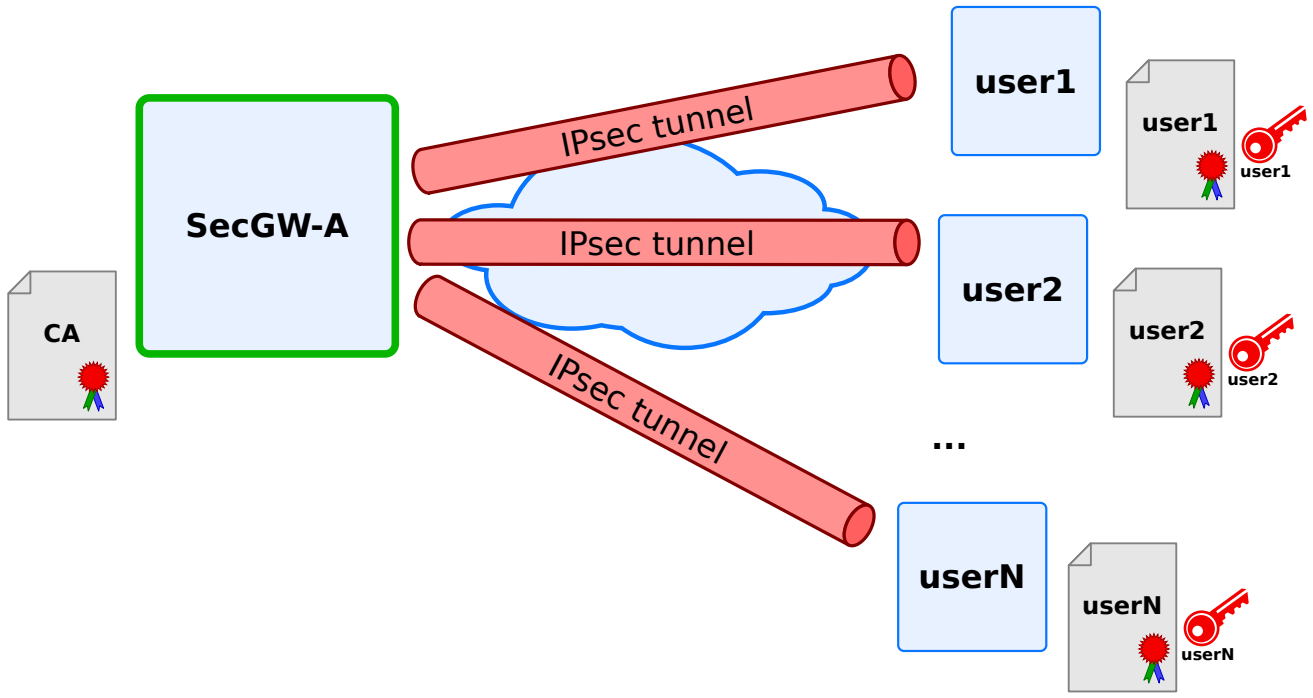
Certificate authentication

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 autority (CA), whose public key is stored in a CA certificate. The CA certificate enables to validate the authenticity of all certificates that it delivered.

Like for bank cards, CAs 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-----
... MIIB9jCCAV8CAQMwDQYJKoZIhvcNAQEEBQAwUzETMBEGA1UEChMKNldJTkQgUy5B
... LjEOMAwGA1UEBxMFUGFyaXMxCzAJBgNVBAYTAkZSMR8wHQYDVQQDExZIZWFkcXVh
... cnRlcnMgQXV0aG9yaXR5MB4XDTE4MDkxOTEzMjM1MloXDTE5MDkxOTEzMjM1Mlow
... NDELMAkGA1UEBhMCRlIxEzARBgNVBAoTCjZXSU5EIFMuQS4xEDAOBgNVBAMTB1Nl
... Y0dXLUEwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAOuCfHphepTnllpX/emq
... IMjW35RAm3TSSHSgDvBm/QtBHgJgLd53ANGbRQ7olinx7jA+CrbrBM9BdEXdR7So
... Q9++munDep/Eb9vu55mMm/leZ8xnV4jIDjLmHCP/AMPNYzKVJHPCElDIbLsbvHIq
... 8A6CYaQOi7NkOrkRY9q3LiEzAgMBAAEwDQYJKoZIhvcNAQEEBQADgYEAdSmnAN5+
... eRh7WuxuAlSGJh1PWb3NzrSKcbJnMPMz1qCqVhvQiGTQNIE5rpr6AlJN7LZV/wvS
```

(continues on next page)

(continued from previous page)

```
... ng4yIizgehU0fluNfAroTEOoxq06m39YZPoY6mUNk82kRq3YTEx+j9EizRjePHzk
... jfYhCQITZa0atkjpfI143bO/k1NVC9exBv0=
... -----END CERTIFICATE-----"
vrouter running certificate secgw-a#! private-key "-----BEGIN PRIVATE KEY-----
... MIICdgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBAOuCfHphepTnllpX
... /emqIMjW35RAm3TSSHSgDvBm/QtBHgJgLd53ANGbRQ7olinx7jA+CrbrBM9BdEXd
... R7SoQ9++munDep/Eb9vu55mMm/leZ8xnV4jIDjLmHCP/AMPNYzKVJHPCElDIbLsb
... vHIq8A6CYaQOi7NkOrkRY9q3LiEzAgMBAAECgYB7IBoiBUqIBNeXXf9ypS5Esgnr
... wSdFGRcmWfPVfZJ3ytB8n3n7n62+5/VfyPuQ7FoBwL3rSc2W6Xp3eCuf6ISquXy8
... zNIB2EY4dzXWpzA9E8+0nZiO8dzFyphM0BFN44pwSazrgD0ZSnXQbxzFBwm5+VvC
... cxSpR/A+53bxDklAIQJBAPnMBvgHdtZATV4rzUN42l//McSGgba1GklICul5rIk/
... GhkGLVLgRaxsJoM3myV7lwA/7jJwXX3ypnJEO2uODXECQQDxW6JTUK5N2/0idS1i
... +Y/cEhgv0c7e3zTvTK3qe5t6Q1A2+1n6mpjk4iRSAfsiEMudnUFIBqbCpyZ1/GeV
... 2JbjAkAPau1fL67BCJT94/w2VuY7mJesxpSI/2KQ9VZfFLh2fCOTOdNgUyFZxA8Y
... eD0mMhue01NTX6YVmP12/gkg2VKxAkAUMkLHDf1H7pykAYImwhNTqv/zIG9bHvpi
... +9uhv24nMPLJZwcEfWNF49Z+NkQ5eYZQThRkXoodx7bkMJbKZzFZAkEA+R+jxmK/
... /XiiT7zizYaWW5x/PQrGvpfOehmlcp11+uO3ILDolNqD7gde98P9Rlc2xXF++K8I
... 3yyFFRutrqwKjw==
... -----END PRIVATE KEY-----"
vrouter running certificate secgw-a# ..
vrouter running ike#
```

```
vrouter running ike# show config nodefault
ike
    (...)
    certificate secgw-a
        certificate "-----BEGIN CERTIFICATE-----
MIIB9jCCAV8CAQMwDQYJKoZIhvcNAQEEBQAwUzETMBEGA1UEChMKNldJTkQgUy5B
LjEOMAwGA1UEBxMFUGFyaXMxCzAJBgNVBAYTAkZSMR8wHQYDVQQDExZIZWFkcXVh
cnRlcnMgQXV0aG9yaXR5MB4XDTE4MDkxOTEzMjM1MloXDTE5MDkxOTEzMjM1Mlow
NDELMAkGA1UEBhMCRlIxEzARBgNVBAoTCjZXSU5EIFMuQS4xEDAOBgNVBAMTB1Nl
Y0dXLUEwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAOuCfHphepTnllpX/emq
IMjW35RAm3TSSHSgDvBm/QtBHgJgLd53ANGbRQ7olinx7jA+CrbrBM9BdEXdR7So
Q9++munDep/Eb9vu55mMm/leZ8xnV4jIDjLmHCP/AMPNYzKVJHPCElDIbLsbvHIq
8A6CYaQOi7NkOrkRY9q3LiEzAgMBAAEwDQYJKoZIhvcNAQEEBQADgYEAdSmnAN5+
eRh7WuxuAlSGJh1PWb3NzrSKcbJnMPMz1qCqVhvQiGTQNIE5rpr6AlJN7LZV/wvS
ng4yIizgehU0fluNfAroTEOoxq06m39YZPoY6mUNk82kRq3YTEx+j9EizRjePHzk
jfYhCQITZa0atkjpfI143bO/k1NVC9exBv0=
-----END CERTIFICATE-----"
        private-key "-----BEGIN PRIVATE KEY-----
MIICdgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBAOuCfHphepTnllpX
/emqIMjW35RAm3TSSHSgDvBm/QtBHgJgLd53ANGbRQ7olinx7jA+CrbrBM9BdEXd
R7SoQ9++munDep/Eb9vu55mMm/leZ8xnV4jIDjLmHCP/AMPNYzKVJHPCElDIbLsb
vHIq8A6CYaQOi7NkOrkRY9q3LiEzAgMBAAECgYB7IBoiBUqIBNeXXf9ypS5Esgnr
wSdFGRcmWfPVfZJ3ytB8n3n7n62+5/VfyPuQ7FoBwL3rSc2W6Xp3eCuf6ISquXy8
zNIB2EY4dzXWpzA9E8+0nZiO8dzFyphM0BFN44pwSazrgD0ZSnXQbxzFBwm5+VvC
cxSpR/A+53bxDklAIQJBAPnMBvgHdtZATV4rzUN42l//McSGgba1GklICul5rIk/
GhkGLVLgRaxsJoM3myV7lwA/7jJwXX3ypnJEO2uODXECQQDxW6JTUK5N2/0idS1i
+Y/cEhgv0c7e3zTvTK3qe5t6Q1A2+1n6mpjk4iRSAfsiEMudnUFIBqbCpyZ1/GeV
```

(continues on next page)

(continued from previous page)

```
2JbjAkAPau1fL67BCJT94/w2VuY7mJesxpSI/2KQ9VZfFLh2fCOTOdNgUyFZxA8Y
eD0mMhue01NTX6YVmP12/gkg2VKxAkAUMkLHDf1H7pykAYImwhNTqv/zIG9bHvpi
+9uhv24nMPLJZwcEfWNF49Z+NkQ5eYZQThRkXoodx7bkMJbKZzFZAkEA+R+jxmK/
/XiiT7zizYaWW5x/PQrGvpfOehmlcp11+uO3ILDolNqD7gde98P9Rlc2xXF++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 overriden 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.

```
vrouter running ike# vpn siteA-roadw
vrouter running vpn siteA-roadw#! ike-policy
vrouter running ike-policy#! template iketemp1
vrouter running ike-policy#! local-auth-method certificate
vrouter running ike-policy#! ..
vrouter running vpn siteA-roadw#! ipsec-policy template ipsectemp1
vrouter running vpn siteA-roadw# certificate secgw-a
vrouter running vpn siteA-roadw# ..
vrouter running ike#
```

```
vrouter running ike# show config
ike
    (...)
    vpn siteA-roadw
        ike-policy
            template iketemp1
            local-auth-method certificate
            ..
        ipsec-policy
            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:

```
vrouter running ike# certificate-authority hq-authority
vrouter running certificate-authority hq-authority# certificate "-----BEGIN␣
↪CERTIFICATE-----
... MIIC2zCCAkSgAwIBAgIJAJpUB7T8zBYBMA0GCSqGSIb3DQEBBAUAMFMxEzARBgNV
... BAoTCjZXSU5EIFMuQS4xDjAMBgNVBAcTBVBhcmlzMQswCQYDVQQGEwJGUjEfMB0G
... A1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhvcml0eTAeFw0xODA5MTkxMzE5MTNaFw0x
... ODEwMTkxMzE5MTNaMFMxEzARBgNVBAoTCjZXSU5EIFMuQS4xDjAMBgNVBAcTBVBh
... cmlzMQswCQYDVQQGEwJGUjEfMB0GA1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhvcml0
... eTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA2mWsQQ14SSkx0Qp5eXXHMkAV
... OEyIJVD3dVPrcQkeCUR38KPrA8Dmlt/KLTrTfat6+/wxS1HywCLYR3U1+CrEQmR+
... kC/NgcNC+QqXyevb+2LTT606oHMQ6XckWIDhhD6JszN0dtcAci1SMgaKIoaoxElu
... TwIdDBkj8W7gnpn84k8CAwEAAaOBtjCBszAMBgNVHRMEBTADAQH/MB0GA1UdDgQW
... BBSN5H+zxbYDk/kVJuqimYsT2oDGDTCBgwYDVR0jBHwweoAUjeR/s8W2A5P5FSbq
... opmLE9qAxg2hV6RVMFMxEzARBgNVBAoTCjZXSU5EIFMuQS4xDjAMBgNVBAcTBVBh
... cmlzMQswCQYDVQQGEwJGUjEfMB0GA1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhvcml0
... eYIJAJpUB7T8zBYBMA0GCSqGSIb3DQEBBAUAA4GBAEvu9Rj1dUcQsFywseZdZcC7
... 9jxhHtml1naxqDp/krPG/GJiSiCypQOGjbcXlRa2NOtLU7DwZTKH3S3fw8TBIAen
... 7vbQFLUtzrZ07TW4wnmtBtGd7GVqAZVIoUnkldVHhHL6hGy2DM+3e8+lptx8+tb6
... U/7s2V3Bm/HkQRq8+Gji
... -----END CERTIFICATE-----"
vrouter running certificate-authority hq-authority# ..
vrouter running ike#
```

```
vrouter running ike# show config nodefault
ike
    (...)
    certificate-authority hq-authority
        certificate "-----BEGIN CERTIFICATE-----
MIIC2zCCAkSgAwIBAgIJAJpUB7T8zBYBMA0GCSqGSIb3DQEBBAUAMFMxEzARBgNV
BAoTCjZXSU5EIFMuQS4xDjAMBgNVBAcTBVBhcmlzMQswCQYDVQQGEwJGUjEfMB0G
A1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhvcml0eTAeFw0xODA5MTkxMzE5MTNaFw0x
ODEwMTkxMzE5MTNaMFMxEzARBgNVBAoTCjZXSU5EIFMuQS4xDjAMBgNVBAcTBVBh
cmlzMQswCQYDVQQGEwJGUjEfMB0GA1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhvcml0
eTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA2mWsQQ14SSkx0Qp5eXXHMkAV
OEyIJVD3dVPrcQkeCUR38KPrA8Dmlt/KLTrTfat6+/wxS1HywCLYR3U1+CrEQmR+
kC/NgcNC+QqXyevb+2LTT606oHMQ6XckWIDhhD6JszN0dtcAci1SMgaKIoaoxElu
TwIdDBkj8W7gnpn84k8CAwEAAaOBtjCBszAMBgNVHRMEBTADAQH/MB0GA1UdDgQW
BBSN5H+zxbYDk/kVJuqimYsT2oDGDTCBgwYDVR0jBHwweoAUjeR/s8W2A5P5FSbq
opmLE9qAxg2hV6RVMFMxEzARBgNVBAoTCjZXSU5EIFMuQS4xDjAMBgNVBAcTBVBh
cmlzMQswCQYDVQQGEwJGUjEfMB0GA1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhvcml0
eYIJAJpUB7T8zBYBMA0GCSqGSIb3DQEBBAUAA4GBAEvu9Rj1dUcQsFywseZdZcC7
9jxhHtml1naxqDp/krPG/GJiSiCypQOGjbcXlRa2NOtLU7DwZTKH3S3fw8TBIAen
7vbQFLUtzrZ07TW4wnmtBtGd7GVqAZVIoUnkldVHhHL6hGy2DM+3e8+lptx8+tb6
U/7s2V3Bm/HkQRq8+Gji
-----END CERTIFICATE-----"
```

(continues on next page)

(continued from previous page)

```
        ..
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 overriden 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 iketemp1
vrouter running ike-policy#! remote-auth-method certificate
vrouter running ike-policy#! ..
vrouter running vpn siteA-roadw#! ipsec-policy template ipsectemp1
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 iketemp1
            remote-auth-method certificate
            ..
        ipsec-policy
            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-----
... MIIByjCCATMCAQEwDQYJKoZIhvcNAQEEBQAwUzETMBEGA1UEChMKNldJTkQgUy5B
... LjEOMAwGA1UEBxMFUGFyaXMxCzAJBgNVBAYTAkZSMR8wHQYDVQQDExZIZWFkcXVh
```

(continues on next page)

(continued from previous page)

```
... cnRlcnMgQXV0aG9yaXR5Fw0xODA5MTkxMzI2MTlaFw0xODEwMTkxMzI2MTlaMBQw
... EgIBARcNMTgwOTE5MTMyMzM0WqCBlTCBkjCBgwYDVR0jBHwweoAUjeR/s8W2A5P5
... FSbqopmLE9qAxg2hV6RVMFMxEzARBgNVBAoTCjZXSU5EIFMuQS4xDjAMBgNVBAcT
... BVBhcmlzMQswCQYDVQQGEwJGUjEfMB0GA1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhv
... cml0eYIJAJpUB7T8zBYBMAoGA1UdFAQDAgEBMA0GCSqGSIb3DQEBBAUAA4GBAAtY
... 3gXNIMwMjH6rafv9wI5qrDCwOp7KNdcrZbNuV/RURJ9mle8EPJ01PJSnxPMuIuzX
... VGbgjRxagWAQLlj4bkhHiqiezThi0D5xTSmmmXEZ52oK5GVDjElWU9OZeK1vssLL
... 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-----
MIIByjCCATMCAQEwDQYJKoZIhvcNAQEEBQAwUzETMBEGA1UEChMKNldJTkQgUy5B
LjEOMAwGA1UEBxMFUGFyaXMxCzAJBgNVBAYTAkZSMR8wHQYDVQQDExZIZWFkcXVh
cnRlcnMgQXV0aG9yaXR5Fw0xODA5MTkxMzI2MTlaFw0xODEwMTkxMzI2MTlaMBQw
EgIBARcNMTgwOTE5MTMyMzM0WqCBlTCBkjCBgwYDVR0jBHwweoAUjeR/s8W2A5P5
FSbqopmLE9qAxg2hV6RVMFMxEzARBgNVBAoTCjZXSU5EIFMuQS4xDjAMBgNVBAcT
BVBhcmlzMQswCQYDVQQGEwJGUjEfMB0GA1UEAxMWSGVhZHF1YXJ0ZXJzIEF1dGhv
cml0eYIJAJpUB7T8zBYBMAoGA1UdFAQDAgEBMA0GCSqGSIb3DQEBBAUAA4GBAAtY
3gXNIMwMjH6rafv9wI5qrDCwOp7KNdcrZbNuV/RURJ9mle8EPJ01PJSnxPMuIuzX
VGbgjRxagWAQLlj4bkhHiqiezThi0D5xTSmmmXEZ52oK5GVDjElWU9OZeK1vssLL
PK9DsxuURw0RP32iv6l68qwaPdI4tR0K8wcVXPn9
-----END X509 CRL-----"
        ..
    ..
```

To add a CRL distribution point, specify the ldap or http URI. CRLs must be encoded in Distinguished Encoding Rules (DER) 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
```

(continues on next page)

(continued from previous page)

```
        ..
    ..
```

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 authentified 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 overriden 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 overriden 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
        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
```

(continues on next page)

(continued from previous page)

```
        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
            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
        invalid 0
        invalid-spi 0
        ike-init-in-req 0
        ike-init-in-resp 1
        ike-init-out-req 1
```

(continues on next page)

(continued from previous page)

```
        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
            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
```

(continues on next page)

(continued from previous page)

```
            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
            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
                ..
            ..
        ..
    ..
```

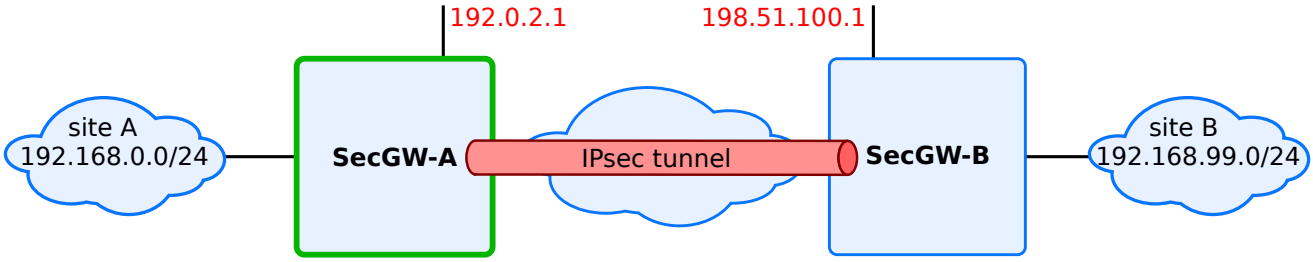
The state dumps:

- 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`).

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
```

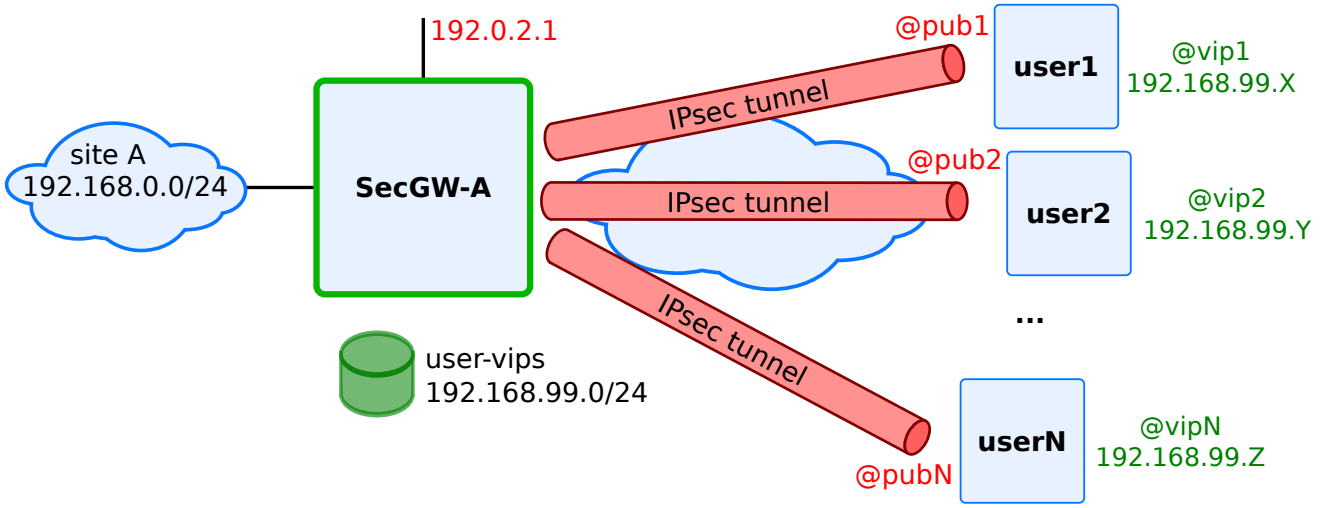
(continues on next page)

(continued from previous page)

```
            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
            ..
        ..
    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
            ..
```

(continues on next page)

(continued from previous page)

```
    ..
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
    ..
```

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
```

(continues on next page)

(continued from previous page)

```
        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):

```
vrouter running ike# ipsec-policy-template ipsectemp1
vrouter running ipsec-policy-template ipsectemp1# esp-proposal 1
vrouter running esp-proposal 1# esn false
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
            esn false
            ..
        ..
    ..
```

To specify that ESN is not mandatory but should be negotiated, specify both `esn true` and `esn false`, by order of preference:

```
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# esn false
vrouter running esp-proposal 1# ..
vrouter running ipsec-policy-template ipsectemp1# ..
```

```
vrouter running ike# show config
ike
    (...)
    ipsec-policy-template ipsectemp1
        esp-proposal 1
            enc-alg aes128-cbc
            auth-alg hmac-sha256
```

(continues on next page)

(continued from previous page)

```
            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
vrouter running pool my-pool#! address 192.168.1.1-192.168.2.127
vrouter running pool my-pool# dns 192.168.3.99
vrouter running pool my-pool# nbns 192.168.3.99
vrouter running pool my-pool# dhcp 192.168.3.100
vrouter running pool my-pool# ..
vrouter 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:

```
vrouter running ike# vpn vpn-secgw
vrouter running vpn vpn-secgw# vip-pool my-pool
vrouter running vpn vpn-secgw# ..
vrouter 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):

```
vrouter running ike# vpn vpn-secgw
vrouter running vpn vpn-secgw# security-policy dynamic-vip
vrouter running security-policy dynamic-vip# local-ts subnet 10.100.0.64/26
vrouter running security-policy dynamic-vip# remote-ts protocol 6
vrouter running security-policy dynamic-vip# ..
vrouter running vpn vpn-secgw# ..
vrouter 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:

$$\text{timeout}_{\text{try}} = \texttt{retransmit-timeout} \times \texttt{retransmit-base}^{\text{try}}$$

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 nodefault
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/Ret

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 nodefault
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 nodefault
ike
    (...)
    global-options
        (...)
        dos-protection
            cookie-threshold 12
            block-threshold 6
            init-limit-half-open 100
            ..
        ..
    ..
```

- `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
```

(continues on next page)

(continued from previous page)

```
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 sergments (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 nodefault
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
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
```

SVTI

Security policies can be associated 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 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.

To associate a security policy to an SVTI interface, specify the `svti-id` of the interface on inbound and outbound policies:

```
vrouter running vpn mytunnel-17# security-policy mytunnel
vrouter running security-policy mytunnel-17# svti-id-in 100
vrouter running security-policy mytunnel-17# svti-id-out 100
```

See *SVTI* for details about creating SVTI interfaces.

See also:

The *command reference* for details.

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.

See also:

The *command reference* for details.

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.

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# firewall 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 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 filter# show state
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
```

(continues on next page)

(continued from previous page)

```
        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
        ..
    address-group trusted
        address 2.2.2.2
        address 4.4.4.4
        ..
    ..
```

The same configuration can be made using this NETCONF XML configuration:

```
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>
            <rule>
              <id>1</id>
              <inbound-interface>
                <name>pub0</name>
              </inbound-interface>
              <action>
                <chain>outside</chain>
              </action>
            </rule>
            <rule>
```

(continues on next page)

(continued from previous page)

```
            <id>2</id>
            <inbound-interface>
              <name>pub1</name>
            </inbound-interface>
            <action>
              <chain>outside</chain>
            </action>
          </rule>
          <rule>
            <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>
```

(continues on next page)

(continued from previous page)

```
            </source>
            <description>allow trusted</description>
            <action>
              <standard>accept</standard>
            </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>
      <address-group>
        <name>trusted</name>
        <address>2.2.2.2</address>
        <address>4.4.4.4</address>
      </address-group>
    </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, *vrrp*) 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 afer 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 IKE

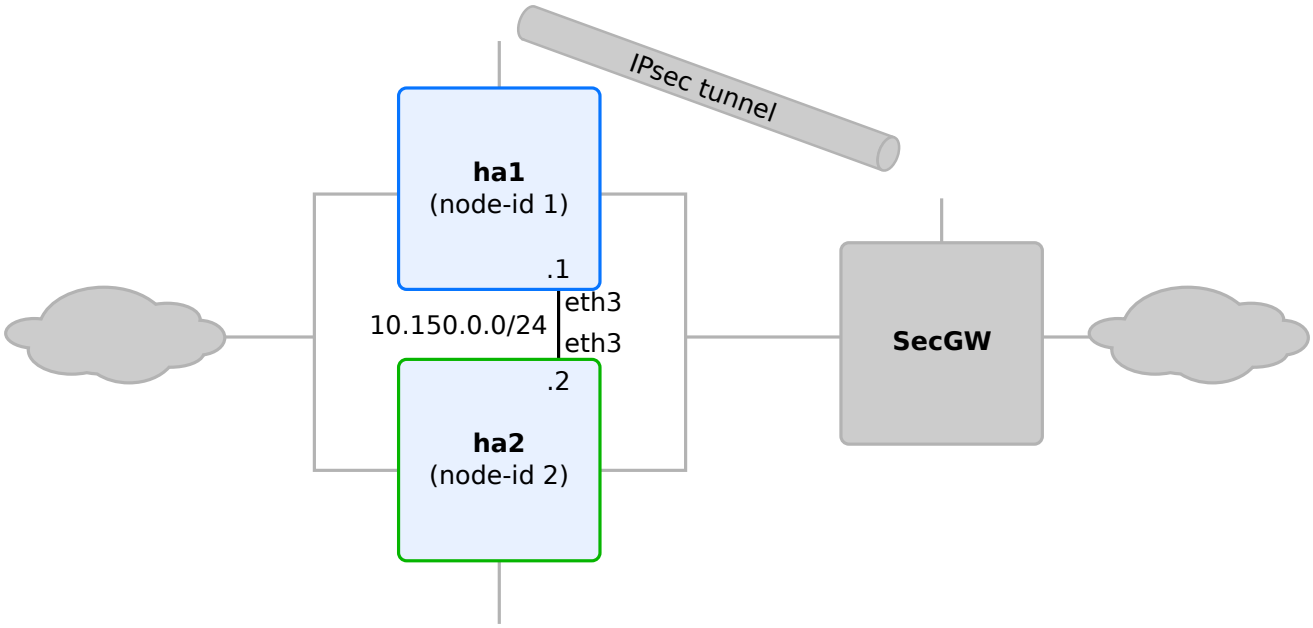
High Availability Internet Key Exchange (HA IKE) is an IKE extension that enables to perform stateful synchronization of IKE between two HA nodes in active/backup mode.

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

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 0 to 15.
- `interface` is the network interface on which syncrhonisation 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.

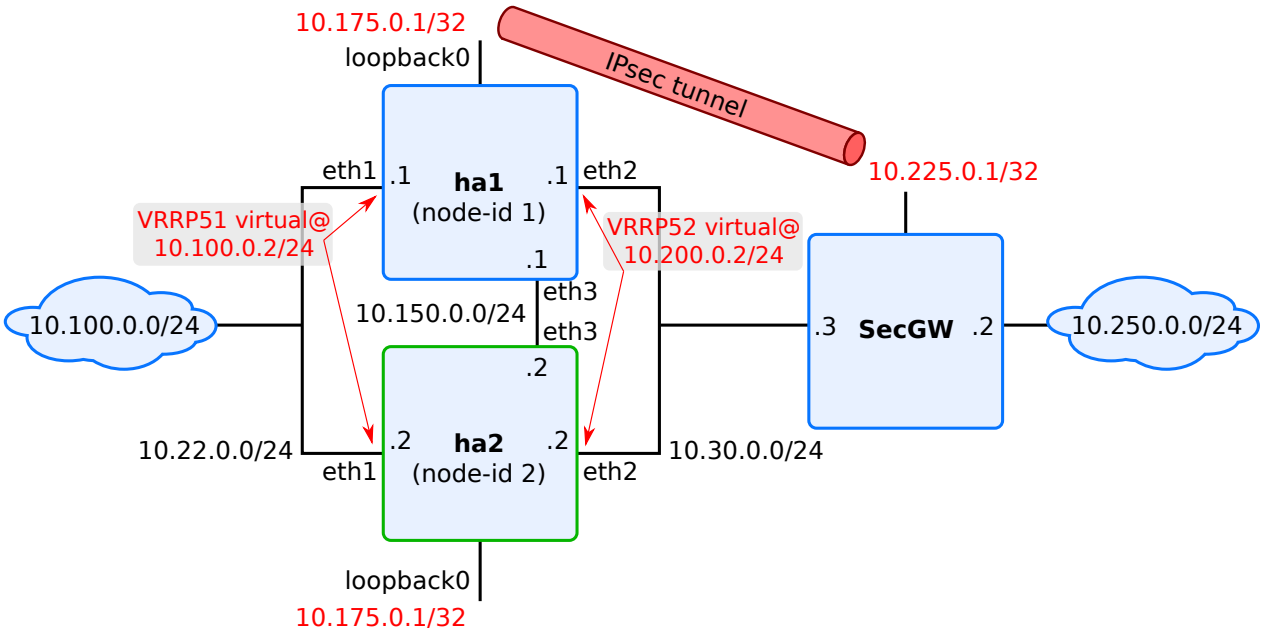
Display HA IKE parameters:

```
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 negotia-

tions 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#! ..
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 nodefault
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
```

(continues on next page)

(continued from previous page)

```
            instance vrrp51
            instance vrrp52
            notify-ha-group ha-group1
            ..
        ..
    ..
```

Configure IKE:

```
ha1 running vrf main# ike
ha1 running ike# ike-policy-template ike1
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
```

(continues on next page)

(continued from previous page)

```
        sp-hash-ipv6
        ..
    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
            ..
        ..
    ..
```

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 nodefault ha
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
                vrid 1
                virtual-address 10.100.0.2/24
                ..
            vrrp vrrp52
```

(continues on next page)

(continued from previous page)

```
            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
                ..
            ..
        ike-policy-template ike1
            ike-proposal 1
                enc-alg aes128-cbc
                auth-alg hmac-sha1
                dh-group modp1024
                ..
            rekey-time 2h
```

(continues on next page)

(continued from previous page)

```
            ..
        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[1] 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.

[1] i.e. the record of the highest SA sequence number of a sent packet protected with this SA

For an input SA, the input sequence number[2] 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).

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:

[2] i.e. the record of the highest SA sequence number of a received packet protected with this SA

```
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 difference between VRRP v2 and VRRP v3 is the support of IPv6 which is only available in the latter. 6WIND VRRP implementation is based on the *Keepalived* open source project.

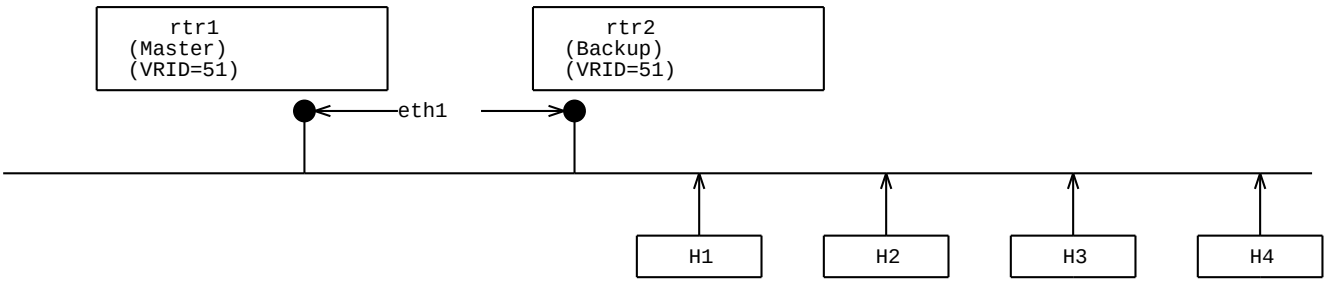
- *Failover groups configuration*
- *High-availability group notification*

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# ..
vrouter running vrf main# interface vrrp vrrp51
vrouter running vrrp vrrp51#! link-interface eth1
vrouter running vrrp vrrp51#! vrid 51
vrouter running vrrp vrrp51# priority 200
vrouter running vrrp vrrp51# virtual-address 10.22.0.1/24
vrouter running vrrp vrrp51# virtual-route 10.22.0.0/24 interface eth1
vrouter running vrrp vrrp51# commit
```

Note: The bound interface *eth1* must be up and have an IP address configured.

Let's fetch the state afer committing this configuration:

```
vrouter running vrf main# vrrp
vrouter running vrrp# show state
vrrp
```

(continues on next page)

(continued from previous page)

```
    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 51
    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:00:01:33
        ..
    virtual-address 10.22.0.1/24
    virtual-route 10.22.0.0/24 interface eth1
    ..
```

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>
```

(continues on next page)

(continued from previous page)

```
    <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>51</vrid>
        <virtual-address>
          <ip>10.22.0.1/24</ip>
        </virtual-address>
        <virtual-route>
          <ip>10.22.0.0/24</ip>
          <interface>eth1</interface>
        </virtual-route>
      </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.

Advertisement interval and preempt delay

`advertisement-interval` specifies the rate at which VRRP advertisements are sent when the router is master.

When the router is backup, it is listening for advertisements. If no advertisement is received, it switches to master after *Master_Down_Interval*. If `preempt` is enabled and advertisements with lower priority are received, if switches to master after `preempt-delay`.

- *Master_Down_Interval*: Time interval for Backup to declare Master down. Calculated as `(3 * advertisement-interval) + Skew_time`.
- *Skew_Time*: Time to skew *Master_Down_Interval* in centiseconds. Calculated as `(((256 - priority) * Master_Adver_Interval) / 256)`

Note: When using slow value for `advertisement-interval`, it is important than the link interface is able to receive packets as soon as the carrier is on, else the `preempt-delay` could be ignored:

- the switch port where the link interface is connected should have its spanning tree disabled.
- if using unicast peering, it is advised to use static ARP entries to avoid latency induced by ARP requests.

Tracking IP addresses

A VRRP instance can track IP addresses. When a tracked address is unreachable, the instance cannot become master.

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*.

Tracking fast path status

A VRRP instance can track the fast path status. If the fast path status does not match the configuration, the instance cannot become master. 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
```

Failover groups configuration

VRRP group is used to group VRRP interfaces from a given VRF that should failover together.

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 51
runnint vrrp vrrp51#! notify-ha-group my-ha-group
runnint vrrp vrrp51# commit
```

Other services support high-availability can declare themselves as subscribers for this group.

3.1.11 Monitoring

KPIs

6WIND KPI (Key Performance Indicator) monitoring provides the ability to monitor and export Turbo IPsec KPIs to an 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/). An example of InfluxDB/Grafana setup is described on 6WIND's github (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) agent to export the specified KPIs to a remote InfluxDB database

To configure the KPIs daemon with everything it can collect, and the `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 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>
```

(continues on next page)

(continued from previous page)

```
      <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>
      <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.

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:

```
vrouter running config# vrf main sflow
vrouter running sflow# sflow-collector 10.0.0.3 port 6343
vrouter running sflow# sflow-interface eth1
vrouter running sflow# sflow-sampling speed 10G rate auto
vrouter running sflow# commit
```

To display the sFlow state:

```
vrouter 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:

```
vrouter> 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>
```

(continues on next page)

(continued from previous page)

```
        <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>
```

See also:

The *command reference* for details.

SNMP

Simple Network Management Protocol (SNMP (Simple Network Management Protocol)) 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.
```

Note: In most cases, you will want to enable the agent in the `main` VRF, but it may be configured in any other VRF.

To display the current SNMP state:

```
vrouter running config# show state vrf main snmp
snmp
    enabled true
```

(continues on next page)

(continued from previous page)

```
    static-info
        location "Copacabana, Rio de Janeiro"
        contact oam@my-company.com
        ..
    community public
        source 10.0.0.0/24
        authorization read-only
        ..
    traps
        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:

```
vrouter 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>
```

(continues on next page)

(continued from previous page)

```
        <authorization>read-only</authorization>
        <source>10.0.0.0/24</source>
      </community>
    </snmp>
  </vrf>
</config>
```

See also:

The *SNMP command reference* for details.

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
vrouter running lldp# interface eth0
vrouter running interface eth0# ..
vrouter running lldp# system-name vrouter
vrouter running lldp# system-description Router
```

(continues on next page)

(continued from previous page)

```
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>
      <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

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.

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>
```

(continues on next page)

(continued from previous page)

```
          <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;
```

(continues on next page)

(continued from previous page)

```
  hardware ethernet de:ed:01:e4:13:29;
}
```

DHCP relay

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
vrouter running dhcp-server 1.0.0.1# interface eth0
vrouter running dhcp-server 1.0.0.1# interface eth1
vrouter running dhcp-server 1.0.0.1# drop-unmatched true
vrouter running dhcp-server 1.0.0.1# ..
vrouter running relay# ..
vrouter running dhcp#
```

The same configuration can be made using this NETCONF XML configuration:

```
vrouter> 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>
```

(continues on next page)

(continued from previous page)

```
        <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.

```
vrouter running config# vrf main
vrouter running vrf main# dns proxy
vrouter running proxy# forward server 192.168.0.254
vrouter running dns# commit
```

To display the DNS proxy state:

```
vrouter 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:

```
vrouter 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>
```

(continues on next page)

(continued from previous page)

```
        </forward>
      </proxy>
    </dns>
  </vrf>
</config>
```

To flush the proxy cache:

```
vrouter> 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 Troubleshooting

Troubleshooting Report

This command allows collecting various diagnostics of the system that can be provided to 6WIND support to help debugging a problem.

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
```

(continues on next page)

(continued from previous page)

```
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
```

(continues on next page)

(continued from previous page)

```
            ..
        partition sda2
            label cloud_Releases
            fstype ext4
            total 9663676416
            available 19223638016
            ..
        partition sda3
            label cloud_Data
            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:

```
vrouter> 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
vrouter> 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-shutdow: 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:

```
vrouter> 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:

```
vrouter> show state network-port
network-port pci-b6s0
    pci-bus-addr 0000:06:00.0
    vendor "Intel Corporation"
    model "I350 Gigabit Network Connection"
    ..
network-port pci-b131s0
    pci-bus-addr 0000:83:00.0
    vendor "Intel Corporation"
    model "82599ES 10-Gigabit SFI/SFP+ Network Connection"
    ..
network-port pci-b131s0f1
    pci-bus-addr 0000:83:00.1
    vendor "Intel Corporation"
    model "82599ES 10-Gigabit SFI/SFP+ Network Connection"
    ..
network-port pci-b134s0
    pci-bus-addr 0000:86:00.0
    vendor "Mellanox Technologies"
    model "MT27700 Family [ConnectX-4]"
    ..
network-port pci-b134s0f1
    pci-bus-addr 0000:86:00.1
    vendor "Mellanox Technologies"
    model "MT27700 Family [ConnectX-4]"
    ..
```

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.

Show Traffic

Display the network traffic flowing through a given network interface.

```
vrouter> cmd show-traffic 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
0 packets dropped by kernel
vrouter>
```

The command can be interrupted by hitting `ctrl-c`.

See also:

The *command reference* for details.

3.1.14 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.

```
vrouter running # system cloud-init
vrouter running cloud-init# enabled false
vrouter running cloud-init# commit
```

To display cloud-init state:

```
vrouter 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:

```
vrouter 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>
```

See also:

The *command reference* for details.

Remote configuration via NETCONF

It is possible to remotely configure the equipement 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 xmltodict 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 xmltodict ncclient

import json
from ncclient import manager
```

(continues on next page)

(continued from previous page)

```
import time
import xmltodict

def connect(host, user, password):
    conn = manager.connect(host=host,
                           username=user,
                           password=password,
                           timeout=10,
                           hostkey_verify=False)

    state = """
<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='replace')

    check_config = conn.validate()
```

(continues on next page)

(continued from previous page)

```
    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)

    print "***************** revert to 'router' ***************"
    send_config = conn.edit_config(target='running', config=conf, default_
↪operation='replace')

    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": {
```

(continues on next page)

(continued from previous page)

```
        "@xmlns": "urn:6wind:vrouter/system",
        "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 IPsec instances.

Dependencies

This guide assumes that you have two (or more) Turbo IPsec 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
$ pip install "ansible > 2.7.10" ncclient jxmlease
...
Successfully installed MarkupSafe-1.1.1 PyYAML-5.1 ansible-2.8.0
```

(continues on next page)

(continued from previous page)

```
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
pycparser-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:
```

(continues on next page)

(continued from previous page)

```
      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

  - 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)
```

(continues on next page)

(continued from previous page)

```
      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}}00/24</ip>
                      </address>
                    </ipv4>
                  </physical>
                </interface>
              </vrf>
            </config>
          </config>

    - name: fetch state again
      netconf_get:
        display: json
        filter: "{{lookup('file', 'filter.xml')}}"
      register: state

    - 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.htm
netconf_config (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) 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>
```

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>
```

(continues on next page)

(continued from previous page)

```
      </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>
```

(continues on next page)

(continued from previous page)

```
    <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) 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 IPsec",
                    "version": "2.2"
                }
            },
            "vrf": {
                "interface": {
                    "physical": [
                        {
                            "ipv4": {
                                "address": {
                                    "ip": "10.0.2.15/24"
```

(continues on next page)

(continued from previous page)

```
                                }
                            },
                            "name": "ens3",
                            "oper-status": "UP",
                            "port": "pci-b0s3"
                        },
                        {
                            "name": "ens4",
                            "oper-status": "DOWN",
                            "port": "pci-b0s4"
                        }
                    ]
                },
                "name": "main"
            }
        }
    }
}
ok: [vrouter1] => {
    "state.output.data": {
        "state": {
            "system": {
                "hostname": "localhost",
                "product": {
                    "license": "valid",
                    "name": "Turbo IPsec",
                    "version": "2.2"
                }
            },
            "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)

```
                "name": "main"
            }
        }
    }
}

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 IPsec",
                    "version": "2.2"
                }
            },
            "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 IPsec",
                    "version": "2.2"
                }
            },
            "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"
                        }
                    ]
                },
                "name": "main"
            }
        }
    }
}
```


(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] ********************
ok: [vrouter1] => {
    "state.output.data": {
        "state": {
            "system": {
                "hostname": "vrouter1",
                "product": {
                    "license": "unknown",
                    "name": "Turbo IPsec",
                    "version": "2.2"
                }
```

(continues on next page)

(continued from previous page)


```
            },
            "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": {
            "system": {
                "hostname": "vrouter2",
                "product": {
                    "license": "unknown",
                    "name": "Turbo IPsec",
                    "version": "2.2"
                }
            },
            "vrf": {
                "interface": {
                    "physical": [
                        {
                            "ipv4": {
                                "address": {
                                    "ip": "10.0.2.15/24"
```


(continues on next page)

(continued from previous page)

```
                                }
                            },
                            "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] ****************************************************
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"
    ]
}
```

(continues on next page)

(continued from previous page)

```
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
```

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]
```

Schedule a system reboot after a grace period.

Input Parameters

delay <uint32> The number of seconds to wait before rebooting. During that time, it is possible to cancel the reboot.

cancel If defined, cancel a pending reboot.

Output Data

reboot-time <string> The time at which the system will reboot.

cmd poweroff

```
vrouter> cmd poweroff [delay <uint32>] [cancel]
```

Schedule a system poweroff after a grace period.

Input Parameters

delay <uint32> The number of seconds to wait before powering off. During that time, it is possible to cancel the poweroff.

cancel If defined, cancel a pending poweroff.

Output Data

poweroff-time <string> The time at which the system will be powered off.

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 <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 <string> 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.

<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 identify-port

```
vrouter> cmd identify-port <port> [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

<port> (mandatory) The port name.

duration <uint16> Length of time to perform the identification, in seconds.

cmd system-image

```
vrouter> cmd system-image install-on-disk <device>
vrouter> cmd system-image import [name <name>] 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>]
```

Manage system images.

Input Parameters

`install-on-disk <device>` Install the system on a specific device.

`<device>` (mandatory) The device on which to install the currently booted image.

`import [name <name>] URL` Import a new system image from a remote URL.

`name <name>` The custom name to assign of the image.

`URL` (mandatory) The URL from which to download the image.

`URL` values	Description
<http[s]://[user:passwd@]host[:port]/path/to/file>	An HTTP(S) file URL.
<sftp://user:passwd@host[:port]/path/to/file>	An SFTP file URL.
<scp://user:passwd@host[:port]/path/to/file>	An SCP file URL.
<ftp://user:passwd@host{[}:port{]}/path/to/file>	An FTP file URL.
<tftp://host{[}:port{]}/path/to/file>	A TFTP file URL.

`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.

cmd license

```
vrouter> cmd license import URL
vrouter> cmd license status
```

Manage license.

Input Parameters

`import URL` Import a license file using an URL.

`URL` (mandatory) The URL from which to download the license.

`URL` values	Description
<http[s]://[user:passwd@]host[:port]/path/to/file>	An HTTP(S) file URL.
<sftp://user:passwd@host[:port]/path/to/file>	An SFTP file URL.
<scp://user:passwd@host[:port]/path/to/file>	An SCP file URL.
<ftp://user:passwd@host{[}:port{]}/path/to/file>	An FTP file URL.
<tftp://host{[}:port{]}/path/to/file>	A TFTP file URL.

`status` Display the license status.

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 url URL <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 url URL <name>` Export an existing troubleshooting report to a remote server via SFTP.

`url URL` (mandatory) The destination URL.

URL values	Description
<sftp://user:passwd@host[:port]/path/to/file>	An SFTP file URL.
<scp://user:passwd@host[:port]/path/to/file>	An SCP file URL.
<smtp[s]://[user:passwd@]host/email.addr@domain.tdl>	An SMTP(S) email URL.
<ftp://user:passwd@host{[}:port{]}/path/to/file>	An FTP file URL.
<tftp://host{[}:port{]}/path/to/file>	A TFTP file URL.
<http[s]://[user:passwd@]host[:port]/path/to/file>	An HTTP(S) file URL.

<name> (mandatory) The name of the report to export.

cmd dns proxy clear-cache

```
vrouter> cmd dns proxy clear-cache [vrf <string>]
```

Clear DNS proxy cache.

Input Parameters

vrf <string> Specify the VRF.

3.2.2 show

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 ipv4-routes

```
vrouter> show ipv4-routes [vrf <string>] [table <uint32>] [to TO]
```

Show IPv4 routing table.

Input Parameters

vrf <string> Specify the VRF.

table <uint32> Non-main Kernel Routing Table.

to TO Find the route entry used to reach an IP address.

TO	An IPv4 address.

show ipv6-routes

```
vrouter> show ipv6-routes [vrf <string>] [table <uint32>]
```

Show IPv6 routing table.

Input Parameters

vrf <string> Specify the VRF.

table <uint32> Non-main Kernel Routing Table.

show mpls table

```
vrouter> show mpls table [<uint32>]
```

Show MPLS table information.

Input Parameters

<uint32> LSP to display information about.

show bgp

```
vrouter> show bgp pbr ipset [set <string>] iptable [chain <string>] [vrf <string>]␣
↪\
...             [vrfs] [summary] [neighbors] neighbor [id ID] [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 ip [VALUE] [bestpath] [multipath] prefix [value VALUE] \
...             [bestpath] [multipath] [longer-prefixes] [cidr-only] [statistics] \
...             [summary] [route-map <string>] multicast ip [VALUE] [bestpath] \
...             [multipath] prefix [value VALUE] [bestpath] [multipath] [longer-
↪prefixes] \
...             [cidr-only] [statistics] [summary] [route-map <string>] labeled-
↪unicast \
...             ip [VALUE] [bestpath] [multipath] prefix [value VALUE] [bestpath] \
...             [multipath] [longer-prefixes] [cidr-only] [statistics] [summary] \
...             [route-map <string>] vpn ip [VALUE] [bestpath] [multipath] prefix \
...             [value VALUE] [bestpath] [multipath] [longer-prefixes] [cidr-only] \
...             [statistics] [summary] [route-map <string>] neighbor [id ID] \
...             [advertised-routes] [dampened-routes] [filtered-routes] [flap-
↪statistics] \
...             [prefix-counts] received [prefix-filter] [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 ip [value VALUE] [bestpath]␣
↪[multipath] \
...             prefix [VALUE] [bestpath] [multipath] [longer-prefixes] [cidr-only]␣
↪\
```

(continues on next page)

(continued from previous page)

```
...             [statistics] [summary] [route-map <string>] multicast ip [value␣
↪VALUE] \
...             [bestpath] [multipath] prefix [VALUE] [bestpath] [multipath] \
...             [longer-prefixes] [cidr-only] [statistics] [summary] [route-map
↪<string>] \
...             labeled-unicast ip [value VALUE] [bestpath] [multipath] prefix \
...             [VALUE] [bestpath] [multipath] [longer-prefixes] [cidr-only] \
...             [statistics] [summary] [route-map <string>] vpn ip [value VALUE] \
...             [bestpath] [multipath] prefix [VALUE] [bestpath] [multipath] \
...             [longer-prefixes] [cidr-only] [statistics] [summary] [route-map
↪<string>] \
...             neighbor [id ID] [advertised-routes] [dampened-routes] [filtered-
↪routes] \
...             [flap-statistics] [prefix-counts] received [prefix-filter]␣
↪[received-routes] \
...             [routes] [neighbors]
```

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.

route-map <string> Display information about this route map.

ipv4 ip [VALUE] [bestpath] [multipath] prefix [value VALUE] [bestpath] [multipath] [l
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] [multi
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 ip [VALUE] [bestpath] [multipath] prefix [value VALUE] [bestpath] [multip
Display information for unicast 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.

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 ip [VALUE] [bestpath] [multipath] prefix [value VALUE] [bestpath] [mult
Display information for multicast 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.

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 ip [VALUE] [bestpath] [multipath] prefix [value VALUE] [bestpath] Display information for labeled unicast 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.

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 ip [VALUE] [bestpath] [multipath] prefix [value VALUE] [bestpath] [multipath]
Display information for VPN 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.

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] [advertised-routes] [dampened-routes] [filtered-routes] [flap-st
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.

prefix-counts Display detailed prefix count information.

received [prefix-filter] Display information received from a BGP neighbor.

prefix-filter Display the prefixlist filter.

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] [l
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] [multi
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 ip [value VALUE] [bestpath] [multipath] prefix [VALUE] [bestpath] [multip
Display information for unicast 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.

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 ip [value VALUE] [bestpath] [multipath] prefix [VALUE] [bestpath] [mult
Display information for multicast 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.

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 ip [value VALUE] [bestpath] [multipath] prefix [VALUE] [bestpath]
Display information for labeled unicast 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.

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 ip [value VALUE] [bestpath] [multipath] prefix [VALUE] [bestpath] [multipath]
Display information for VPN 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.

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] [advertised-routes] [dampened-routes] [filtered-routes] [flap-st
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.

prefix-counts Display detailed prefix count information.

received [prefix-filter] Display information received from a BGP neighbor.

prefix-filter Display the prefixlist filter.

received-routes Display the received routes from neighbor.

routes Display routes learned from neighbor.

neighbors Display information about all BGP neighbors.

show ospf

```
vrouter> show ospf [vrf <string>] [route] database [default] router [ADDRESS]␣
↪[neighbor] \
...             interface [NAME]
```

Show OSPF information.

Input Parameters

vrf <string> Specify the VRF.

route OSPF routing table.

database [default] router [ADDRESS] Database summary.

default Database summary.

router [ADDRESS] Database Router link states.

ADDRESS The router address.

ADDRESS values	Description
<A.B.C.D>	An IPv4 address.
self-originate	Self-originated link states.

neighbor Neighbor list.

interface [NAME] Interface information.

NAME The interface name. If not specified, show all interfaces.

NAME	An interface name.

show rip

```
vrouter> show rip [vrf <string>] [status]
```

Show RIP information.

Input Parameters

vrf <string> Specify the VRF.

status Show RIP status.

show ospf6

```
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

```
vrouter> show ripng [status]
```

Show RIPng information.

Input Parameters

status Show RIPng status.

show mpls ldp

```
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>] [ne
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

```
vrouter> 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

```
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 log

```
vrouter> show log [max-lines <uint16>] [service <identityref>] [vrf <string>]␣
↪[facility FACILITY] \
...             level [greater-or-equal GREATER-OR-EQUAL] not
```

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 [greater-or-equal GREATER-OR-EQUAL] not Filter logs by level.

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 Select levels to not show.

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

```
vrouter> show dhcp-server [vrf <string>]
```

Show DHCP server leases.

Input Parameters

vrf <string> Specify the VRF.

show conntracks

```
vrouter> show conntracks [vrf <string>] [family FAMILY] [protocol PROTOCOL]
```

Show conntracks.

Input Parameters

vrf <string> The VRF in which to show the conntracks.

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 ike

```
vrouter> show ike [vrf <string>] 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 <string> Show objects in selected netns only.

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 STAT
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.

3.2.3 flush

flush bgp

```
vrouter> 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] \
```

(continues on next page)

(continued from previous page)

```
...             [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>] [
Flush information about BGP IPv4.

unicast [as AS] [all] [neighbor NEIGHBOR] [external] [neighbor-group <string>] [s
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>]
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 <str
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>] [
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
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.

ipv6 unicast [as AS] [all] [neighbor NEIGHBOR] [external] [neighbor-group <string>] [
Flush information about BGP IPv6.

unicast [as AS] [all] [neighbor NEIGHBOR] [external] [neighbor-group <string>] [s
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>]
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 <str
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>] [
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
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

```
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

```
vrouter> flush ospf6 interface <ifname>
```

Flush OSPFv3 information.

Input Parameters

interface <ifname> (mandatory) The name of the network interface to be cleared.

flush ike ike-sa

```
vrouter> flush ike ike-sa [vrf <string>] [vpn <string>] [unique-id <string>]
```

Flush IKE SA.

Input Parameters

vrf <string> Flush objects in selected netns only.

vpn <string> Flush SA for selected VPN.

unique-id <string> Flush SA with this unique id.

flush ike child-sa

```
vrouter> flush ike child-sa [vrf <string>] [name <string>] [unique-id <string>]
```

Flush child SA.

Input Parameters

vrf <string> Flush objects in selected netns only.

name <string> Flush SA with this name.

unique-id <string> Flush SA with this unique id.

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
```

HOST-NAME	The domain-name type represents a DNS domain name. Fully quallified 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

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
GMT
Antarctica/McMurdo
Antarctica/South_Pole
Antarctica/Rothera

TIMEZONE values
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
America/Maceio
America/Bahia
America/Sao_Paulo
America/Campo_Grande

TIMEZONE values
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
America/Santiago
Pacific/Easter
Asia/Shanghai
Asia/Harbin

TIMEZONE values
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
America/Mazatlan
America/Chihuahua
America/Ojinaga
America/Hermosillo

TIMEZONE values
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
Pacific/Wake
America/New_York
America/Detroit
America/Kentucky/Louisville

TIMEZONE values
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
```

neighbor (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-05-07	20q1	The neighbor advanced configuration is moved in the network-stack container.	/vrouter-system:system/network-stack/neighbor

Neighbor advanced configuration (deprecated).

```
vrouter running config# system neighbor
```

ipv4-max-entries (deprecated)

Maximum number of IPv4 neighbors.

```
vrouter running config# system neighbor
vrouter running neighbor# ipv4-max-entries <uint32>
```

ipv6-max-entries (deprecated)

Maximum number of IPv6 neighbors.

```
vrouter running config# system neighbor
vrouter running neighbor# ipv6-max-entries <uint32>
```

conntrack (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-05-07	20q1	The conntrack advanced configuration is moved in the network-stack container.	/vrouter-system:system/network-stack/conntrack

Conntrack advanced configuration (deprecated).

```
vrouter running config# system conntrack
```

max-entries (deprecated)

Maximum number of Netfilter conntracks.

```
vrouter running config# system conntrack
vrouter running conntrack# max-entries <uint32>
```

tcp-timeout-close (deprecated)

Conntrack TCP timeout close.

```
vrouter running config# system conntrack
vrouter running conntrack# tcp-timeout-close <uint32>
```

tcp-timeout-close-wait (deprecated)

Conntrack TCP timeout close wait.

```
vrouter running config# system conntrack
vrouter running conntrack# tcp-timeout-close-wait <uint32>
```

tcp-timeout-established (deprecated)

Conntrack TCP timeout established.

```
vrouter running config# system conntrack
vrouter running conntrack# tcp-timeout-established <uint32>
```

tcp-timeout-fin-wait (deprecated)

Conntrack TCP timeout fin wait.

```
vrouter running config# system conntrack
vrouter running conntrack# tcp-timeout-fin-wait <uint32>
```

tcp-timeout-last-ack (deprecated)

Conntrack TCP timeout last ack.

```
vrouter running config# system conntrack
vrouter running conntrack# tcp-timeout-last-ack <uint32>
```

tcp-timeout-max-retrans (deprecated)

Conntrack TCP timeout max retrans.

```
vrouter running config# system conntrack
vrouter running conntrack# tcp-timeout-max-retrans <uint32>
```

tcp-timeout-syn-recv (deprecated)

Conntrack TCP timeout syn recv.

```
vrouter running config# system conntrack
vrouter running conntrack# tcp-timeout-syn-recv <uint32>
```

tcp-timeout-syn-sent (deprecated)

Conntrack TCP timeout syn sent.

```
vrouter running config# system conntrack
vrouter running conntrack# tcp-timeout-syn-sent <uint32>
```

tcp-timeout-time-wait (deprecated)

Conntrack TCP timeout time wait.

```
vrouter running config# system conntrack
vrouter running conntrack# tcp-timeout-time-wait <uint32>
```

tcp-timeout-unacknowledged (deprecated)

Conntrack TCP timeout unacknowledged.

```
vrouter running config# system conntrack
vrouter running conntrack# tcp-timeout-unacknowledged <uint32>
```

udp-timeout (deprecated)

Conntrack UDP timeout.

```
vrouter running config# system conntrack
vrouter running conntrack# udp-timeout <uint32>
```

udp-timeout-stream (deprecated)

Conntrack UDP timeout stream.

```
vrouter running config# system conntrack
vrouter running conntrack# udp-timeout-stream <uint32>
```

network-stack

Network stack parameters.

```
vrouter running config# system network-stack
```

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
```

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
```

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
```

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>
```

conntrack

Conntrack advanced configuration.

```
vrouter running config# system network-stack conntrack
```

max-entries

Maximum number of Netfilter conntracks.

```
vrouter running config# system network-stack conntrack
vrouter running conntrack# 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.

name (state only)

The name of the image if it is defined.

```
vrouter> show state system installed-image <string> name
```

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 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>
```

<string>	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
```

PASSWORD	The crypt-hash type is used to store passwords using a hash function. The algorithms for applying the 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>
```

3.2.7 aaa

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
```

3.2.8 vrf

Vrf list.

```
vrouter running config# vrf <vrf>
```

<vrf> values	Description
main	The main vrf.
<string>	The vrf name.

3.2.9 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
```

3.2.10 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 quallified 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-TO
```

LISTEN-TO 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
```

3.2.11 lldp

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
```

3.2.12 kpi

KPI configuration for interface and telegraf agent.

```
vrouter running config# vrf <vrf> kpi
```

enabled (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-01-22	19q3	There is only one kpi daemon, which runs in the main vrf. It makes more sense to configure it in system.	/vrouter-system:system/vrouter-kpi:kpi/enabled

Enable or disable the KPIs.

```
vrouter running config# vrf <vrf> kpi
vrouter running kpi# enabled true|false
```

service (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-01-22	19q3	There is only one kpi daemon, which runs in the main vrf. It makes more sense to configure it in system.	/vrouter-system:system/vrouter-kpi:kpi/service

The list of activated services. Default is all.

```
vrouter running config# vrf <vrf> kpi
vrouter running kpi# service SERVICE
```

SERVICE values	Description
fp-bridge-stats	Fast path bridge statistics.
fp-context-switch-stats	Fast path cores context switch statistics.
fp-cp-protect-stats	Fast path control plane protection statistics.
fp-cpu-usage	Fast path cpu usage.
fp-dpvi-stats	Fast path dataplane virtual interafce statistics.
fp-ebtables-stats	Fast path ethernet filtering statistics.
fp-exception-queue-stats	Fast path exception queues statistics.
fp-exceptions-stats	Fast path exceptions statistics.
fp-filling	Fast path tables filling.
fp-filling-cg-nat	Fast path tables filling for cg-nat module.
fp-global-stats	Fast path global statistics.
fp-gre-stats	Fast path GRE statistics.

continues on next page

Table 2 – continued from previous page

SERVICE values	Description
fp-gro-stats	Fast path GRO statistics.
fp-ip-stats	Fast path IPv4 statistics.
fp-ip6-stats	Fast path IPv6 statistics.
fp-ipsec-stats	Fast path IPSec statistics.
fp-ipsec6-stats	Fast path IPSecv6 statistics.
fp-npf-stats	Fast path standalone filtering statistics (deprecated, use fp-cg-nat-stats instead).
fp-cg-nat-stats	Fast path CG-NAT statistics.
fp-ports-stats	Fast path ports statistics.
fp-status	Fast path status.
fp-vlan-stats	Fast path VLAN statistics.
fp-vxlan-stats	Fast path VXLAN statistics.
network-nic-eth-stats	NICs hardware counters.
network-nic-hw-info	NICs hardware informations.
network-nic-traffic-stats	NICs traffic statistics.
product-license	Product license status.
product-version	Product version.
system-cpu-usage	Operating system cpu usage.
system-disk-usage	Operating system disk usage.
system-memory	Operating system memory usage.
system-numa-stats	Operating system NUMA usage.
system-processes	Operating system process list and load.
system-soft-interrupts-stats	Operating system software interrupts.
system-uptime	Operating system uptime.
system-user-count	Operating system user currently logged count.
system-users	Operating system user currently logged list.

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.

3.2.13 telegraf

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.
<http[s]://host[:port]>	An 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
```

3.2.14 tracker

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	Ths 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
```

negociated (state only)

BFD negociated operational state data.

transmission-interval (state only)

Negotiated transmit interval.

```
vrouter> show state tracker bfd <bfd> negociated transmission-interval
```

receive-interval (state only)

Negotiated receive interval.

```
vrouter> show state tracker bfd <bfd> negociated receive-interval
```

echo-transmission-interval (state only)

Negotiated echo transmit interval.

```
vrouter> show state tracker bfd <bfd> negociated echo-transmission-interval
```

icmp

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 quallified 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>	An IPv4 address.
<X:X::X:X>	An IPv6 address.
<host-name>	The domain-name type represents a DNS domain name. Fully quallified 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
```

3.2.15 nat

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 \
...   source \
...     address [not] VALUE \
...     port [not] VALUE \
...   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
```

address

Match this destination address or prefix.

```
address [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

The address to match.

```
VALUE
```

VALUE values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	A IPv4 address.
<A.B.C.D/M>	A IPv4 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.

source

Match a source attribute.

```
source \
     address [not] VALUE \
     port [not] VALUE
```

address

Match this source address or prefix.

```
address [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

The address to match.

```
VALUE
```

VALUE values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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.

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 \
...   source \
...     address [not] VALUE \
...     port [not] VALUE \
...   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
```

address

Match this destination address or prefix.

```
address [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

The address to match.

```
VALUE
```

VALUE values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An IPv4 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.

source

Match a source attribute.

```
source \
     address [not] VALUE \
     port [not] VALUE
```

address

Match this source address or prefix.

```
address [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

The address to match.

```
VALUE
```

VALUE values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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.

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 \
```

(continues on next page)

(continued from previous page)

```
    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
```

3.2.16 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>
```

<uint16>	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>
```

<server> 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 quallified 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
```

3.2.17 firewall

ipv4 filter

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.

```
vrouter running config# vrf <vrf> firewall ipv4 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> \
...   ipv4 [not] fragment \
...   icmp-type [not] VALUE \
...   tcp-flags [not] set SET examined EXAMINED \
...   conntrack \
```

(continues on next page)

(continued from previous page)

```
...     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 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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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.
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 3 – 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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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 <leafref> 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
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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> \
...   source \
```

(continues on next page)

(continued from previous page)

```
...     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 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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	A IPv4 address.
<A.B.C.D/M>	A 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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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 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
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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 \
...   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 \
```

(continues on next page)

(continued from previous page)

```
...     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 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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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.
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.

continues on next page

Table 5 – continued from previous page

`VALUE` values	Description
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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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).

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 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
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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> \
...   source \
```

(continues on next page)

(continued from previous page)

```
...     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 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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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.

continues on next page

Table 6 – continued from previous page

VALUE values	Description
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.
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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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 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>
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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
```

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

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 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 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 \
...     connmark \
...       set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...       save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...       restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
```

(continues on next page)

(continued from previous page)

```
...     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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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.
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.

continues on next page

Table 7 – continued from previous page

`VALUE` values	Description
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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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>
```

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 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>
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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
```

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 \
...   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> dscp DSCP \
...     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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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.

continues on next page

Table 8 – continued from previous page

VALUE values	Description
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.
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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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 <leafref> dscp DSCP \
     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>
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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
```

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 \
```

(continues on next page)

(continued from previous page)

```
...     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> dscp DSCP \
...     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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	A IPv4 address.
<A.B.C.D/M>	A 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.
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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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> dscp DSCP \
     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>
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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
```

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 \
...   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 \
```

(continues on next page)

(continued from previous page)

```
...     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 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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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.
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.

continues on next page

Table 10 – continued from previous page

`VALUE` values	Description
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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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).

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 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>
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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
```

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 \
...   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 \
```

(continues on next page)

(continued from previous page)

```
...     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 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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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.
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.

continues on next page

Table 11 – continued from previous page

`VALUE` values	Description
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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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).

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 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>
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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
```

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 \
...   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 <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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	A IPv4 address.
<A.B.C.D/M>	A 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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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 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>
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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
```

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

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 \
...     connmark \
...       set-xmark <0x0-0xffffffff> mask <0x0-0xffffffff> \
...       save-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
...       restore-mark nfmask <0x0-0xffffffff> ctmask <0x0-0xffffffff> \
```

(continues on next page)

(continued from previous page)

```
...     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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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.
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.

continues on next page

Table 13 – continued from previous page

`VALUE` values	Description
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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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>
```

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 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
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 \
```

(continues on next page)

(continued from previous page)

```
...     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 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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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).

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 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
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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 \
...   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 <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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	A IPv4 address.
<A.B.C.D/M>	A 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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 address.
<A.B.C.D/M>	An 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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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 15 – 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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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 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>
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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
```

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

Address group.

```
vrouter running config# vrf <vrf> firewall ipv4 address-group <string>
```

<string>	Name of the address group.

address

List of addresses of the group.

```
vrouter running config# vrf <vrf> firewall ipv4 address-group <string>
vrouter running address-group <string># address ADDRESS
```

ADDRESS	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

Network group.

```
vrouter running config# vrf <vrf> firewall ipv4 network-group <string>
```

<string>	Name of the network group.

network

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 \
...   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 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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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 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
```

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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 \
...   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 <string> 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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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 <string> 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
```

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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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 \
...   asconf-ack forward-tsn \
...     data examined EXAMINED set SET \
...     abort examined EXAMINED set SET \
...     shutdown-complete examined EXAMINED set SET \
```

(continues on next page)

(continued from previous page)

```
...   outbound-interface [not] <string> \
...   action STANDARD chain <string> 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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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).

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 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
```

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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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> \
...   source \
```

(continues on next page)

(continued from previous page)

```
...     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 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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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 <string> 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
```

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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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 \
...   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 <string> \
...     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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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>
```

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 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
```


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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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> \
...   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> \
...     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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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> \
     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
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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 \
...   asconf-ack forward-tsn \
...     data examined EXAMINED set SET \
...     abort examined EXAMINED set SET \
...     shutdown-complete examined EXAMINED set SET \
```

(continues on next page)

(continued from previous page)

```
...   inbound-interface [not] <string> \
...   outbound-interface [not] <string> \
...   action STANDARD chain <string> \
...     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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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 <string> \
     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
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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.

```
vrouter running config# vrf <vrf> firewall ipv6 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> \
...   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> \
...     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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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).

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 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
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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 \
...   asconf-ack forward-tsn \
...     data examined EXAMINED set SET \
...     abort examined EXAMINED set SET \
...     shutdown-complete examined EXAMINED set SET \
```

(continues on next page)

(continued from previous page)

```
...   outbound-interface [not] <string> \
...   action STANDARD chain <string> \
...     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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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).

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 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
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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 \
...   asconf-ack forward-tsn \
...     data examined EXAMINED set SET \
...     abort examined EXAMINED set SET \
...     shutdown-complete examined EXAMINED set SET \
```

(continues on next page)

(continued from previous page)

```
...   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 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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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 <string> 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
```

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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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 \
...   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 <string> notrack \
...     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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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>
```

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 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
```


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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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.

```
vrouter running config# vrf <vrf> firewall ipv6 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> \
...   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 \
```

(continues on next page)

(continued from previous page)

```
...   outbound-interface [not] <string> \
...   action STANDARD chain <string> notrack \
...     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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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).

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 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
```

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 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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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 \
...   asconf-ack forward-tsn \
```

(continues on next page)

(continued from previous page)

```
...     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 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
```

<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 from /etc/protocols.
<string>	Protocol from /etc/protocols.

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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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 values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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/M>	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[,portl,port-port]).

```
port-range [not] VALUE
```

not

Invert the match.

```
not
```

VALUE (mandatory)

Port range, syntax is port[,portl,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.

conntrack

Match conntrack information.

```
conntrack \
     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.

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 <string> 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
```

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 (nfmark) to the connection mark (ctmark) using the given masks. The new value is determined as follows: ctmark = (ctmark & ~ctmask) ^ (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 & ~nfmask) ^ (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 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

Address group.

```
vrouter running config# vrf <vrf> firewall ipv6 address-group <string>
```

<string>	Name of the address group.

address

List of addresses of the group.

```
vrouter running config# vrf <vrf> firewall ipv6 address-group <string>
vrouter running address-group <string># address ADDRESS
```

ADDRESS	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

Network group.

```
vrouter running config# vrf <vrf> firewall ipv6 network-group <string>
```

<string>	Name of the network group.

network

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.

3.2.18 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
```

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
```

3.2.19 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
```

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>	An IPv4 prefix: address and CIDR mask.

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>	An IPv6 prefix: address and CIDR mask.

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.

```
vrouter running config# vrf <vrf> interface bridge <bridge> 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 bridge <bridge> ipv6 neighbor <neighbor>␣
↪router
```

state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface bridge <bridge> ipv6 neighbor <neighbor>␣
↪state
```

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>
```

<link-interface>	An interface name.

qos

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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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

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
```

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 (mandatory)

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.

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 <leafref>
```

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>	An IPv4 prefix: address and CIDR mask.

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
```

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>	An IPv6 prefix: address and CIDR mask.

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
```

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

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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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

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 <leafref>
```

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>	An IPv4 prefix: address and CIDR mask.

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>	An IPv6 prefix: address and CIDR mask.

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
```

qos

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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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.

```
vrouter> 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.

```
vrouter> 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).

```
vrouter> 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).

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> show state vrf <vrf> interface ipip <ipip> counters out-errors
```

lag

The list of LAG interfaces on the device.

```
vrouter 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>	An IPv4 prefix: address and CIDR mask.

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.

```
vrouter running config# vrf <vrf> interface lag <lag> ipv6
vrouter running ipv6# address <address> peer PEER
```

<address> values	Description
<X:X::X:X>	An IPv6 address.
<X:X::X:X/M>	An IPv6 prefix: address and CIDR mask.

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 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.

```
vrouter> show state vrf <vrf> interface lag <lag> ipv6 address <address> status
```

neighbor

List of IPv6 neighbors.

```
vrouter running config# vrf <vrf> interface lag <lag> 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 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
```

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.

qos

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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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

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>	An IPv4 prefix: address and CIDR mask.

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>	An IPv6 prefix: address and CIDR mask.

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
```

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

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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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'.

```
vrouter> 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'.

```
vrouter> show state vrf <vrf> interface loopback <loopback> counters out-errors
```


physical

The list of physical interfaces on the device.

```
vrouter running config# vrf <vrf> interface physical <physical>
```

<physical>	An interface name.

port (mandatory)

Reference to a physical network port.

```
vrouter running config# vrf <vrf> interface physical <physical>
vrouter running physical <physical># port <port>
```

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>	An IPv4 prefix: address and CIDR mask.

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>	An IPv6 prefix: address and CIDR mask.

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
```


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	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 unpopuplated (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

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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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.

```
vrouter> 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.

```
vrouter> 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.

```
vrouter> 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.

```
vrouter> 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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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.

```
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 <0x0-0xffffffff> 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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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.

```
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 <0x0-0xffffffff> 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-dwrr 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'.

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> 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

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>	An IPv4 prefix: address and CIDR mask.

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 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.

```
vrouter running config# vrf <vrf> interface svti <svti> 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 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>	An IPv6 prefix: address and CIDR mask.

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.

```
vrouter running config# vrf <vrf> interface svti <svti> 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 svti <svti> ipv6 neighbor <neighbor> router
```

state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface svti <svti> ipv6 neighbor <neighbor> state
```

qos

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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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.

```
vrouter> 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.

```
vrouter> 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).

```
vrouter> 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).

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> 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
```

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.

```
vrouter> 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.

```
vrouter> 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.

```
vrouter> 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).

```
vrouter> show state vrf <vrf> interface veth <veth> last-change
```

ipv4

Parameters for the IPv4 address family.

```
vrouter 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>	An IPv4 prefix: address and CIDR mask.

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>	An IPv6 prefix: address and CIDR mask.

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
```

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

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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
vrouter> show state vrf <vrf> interface veth <veth> qos egress rate-limit policer␣
↪excess-burst
```

shared-policer (state only)

Shared policer name.

```
vrouter> 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.

```
vrouter> 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.

```
vrouter> 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.

```
vrouter> 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.

```
vrouter> 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>	An IPv4 prefix: address and CIDR mask.

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>	An IPv6 prefix: address and CIDR mask.

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
```

qos

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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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'.

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> 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'.

```
vrouter> 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

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 <leafref>
```

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>	An IPv4 prefix: address and CIDR mask.

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 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.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> 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 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>	An IPv6 prefix: address and CIDR mask.

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.

```
vrouter running config# vrf <vrf> interface vxlan <vxlan> 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 vxlan <vxlan> ipv6 neighbor <neighbor>␣
↪router
```

state (state only)

The state of this neighbor entry.

```
vrouter> show state vrf <vrf> interface vxlan <vxlan> ipv6 neighbor <neighbor>␣
↪state
```

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

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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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

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

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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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
```

3.2.20 qos

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-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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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.

Default value

```
1500
```

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-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. 0 allows no excess traffic.

```
vrouter running config# qos policer <string>
vrouter running policer <string># excess-burst EXCESS-BURST
```

EXCESS-BURST	Burst size in bytes. K/M/G/T multipliers are supported. Example: 2K stands for 2000 bytes.

Default value

```
1500
```

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 regular traffic, a.k.a. CBS (Committed Burst Size), in bytes. 0 allows no regular traffic.

```
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. 0 allows no excess traffic.

```
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-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.

```
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.

Default value

```
48000
```

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 automaticaly 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) (mandatory)

Class mark.

```
vrouter running config# qos class <string>
vrouter running class <string># mark <0x0-0xffffffff>
```

3.2.21 vrrp

global

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
```

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

Notify a high-availability group when the group state changes.

```
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

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 = IPv4, 3 = 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 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 a vmac interface for this VRRP instance.

```
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.

```
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, preempt an already running VRRP instance when coming online with a higher priority. 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

Delay the higher priority router waits before preempting.

```
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-interface

List of tracked interfaces. The VRRP instance loses its master state if one of the tracked interfaces go 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 loses its master state if the tracked address is unreachable.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># track TRACK
```

TRACK	An tracker name.

track-fast-path

Prevent the VRRP instance to be master when 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

Notify a high-availability group when the group state changes.

```
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
```

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>
```

track-ip (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-06-07	20q1	IP tracking is now done using generic tracker objects, listed in /vrouter:config/vrouter-tracker:tracker.	/vrouter:config/vrouter:vrf/vrouter-interface:interface/vrouter-vrrp:vrrp/vrouter-vrrp:track

List of tracked addresses. The VRRP instance loses its master state if one of the address is not reachable with a ping.

```
vrouter running config# vrf <vrf> interface vrrp <vrrp>
vrouter running vrrp <vrrp># track-ip <track-ip> vrf VRF period <uint16> \
... threshold <uint8> total <uint8>
```

<track-ip> 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 quallified 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 (deprecated)

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.

period (deprecated)

Time between each ping.

```
period <uint16>
```

threshold (deprecated)

Number of successful pings among <total> to consider peer as reachable.

```
threshold <uint8>
```

total (deprecated)

Check the threshold among this number of last pings to consider peer as reachable.

```
total <uint8>
```

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>	An IPv4 prefix: address and CIDR mask.
<X:X::X:X/M>	An IPv6 prefix: address and CIDR mask.

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
```

3.2.22 ike

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.

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 quallified 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.

logging

Logs configuration.

```
vrouter running config# vrf <vrf> ike logging
```

syslog (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-10-10	19q3	The syslog container has been removed for more readability, and consistency with other services.	None

Logs configuration for each syslog facility.

```
vrouter running config# vrf <vrf> ike logging syslog
```

daemon (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-10-10	19q3	The syslog container has been removed for more readability, and consistency with other services.	/vrouter:vrf/vrouter-ike:ike/vrouter-ike:logging/vrouter-ike:daemon

Max level of messages logged in the system daemons facility.

```
vrouter running config# vrf <vrf> ike logging syslog daemon
```

default (deprecated)

Default max log level.

```
vrouter running config# vrf <vrf> ike logging syslog 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.

asn1 (deprecated)

Low-level encoding/decoding (ASN.1, X.509 etc.).

```
vrouter running config# vrf <vrf> ike logging syslog 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 (deprecated)

Configuration management and plugins.

```
vrouter running config# vrf <vrf> ike logging syslog 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 (deprecated)

CHILD_SA/IPsec SA processing.

```
vrouter running config# vrf <vrf> ike logging syslog 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 (deprecated)

Main daemon setup/cleanup/signal handling.

```
vrouter running config# vrf <vrf> ike logging syslog 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 (deprecated)

Packet encoding/decoding encryption/decryption operations.

```
vrouter running config# vrf <vrf> ike logging syslog 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 (deprecated)

Libipsec library messages.

```
vrouter running config# vrf <vrf> ike logging syslog 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 (deprecated)

IKE_SA/ISAKMP SA processing.

```
vrouter running config# vrf <vrf> ike logging syslog 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 (deprecated)

Jobs queuing/processing and thread pool management.

```
vrouter running config# vrf <vrf> ike logging syslog 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 (deprecated)

IPsec/Networking kernel interface.

```
vrouter running config# vrf <vrf> ike logging syslog 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 (deprecated)

Libstrongwan library messages.

```
vrouter running config# vrf <vrf> ike logging syslog 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 (deprecated)

IKE_SA manager, handling synchronization for IKE_SA access.

```
vrouter running config# vrf <vrf> ike logging syslog 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 (deprecated)

IKE network communication.

```
vrouter running config# vrf <vrf> ike logging syslog 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.

private-authorization (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-10-10	19q3	The syslog container has been removed for more readability, and consistency with other services. The container has been renamed into authpriv.	/vrouter:vrf/vrouter-ike:ike/vrouter-ike:logging/vrouter-ike:authpriv

Max level of messages logged in the private security/authorization messages facility.

```
vrouter running config# vrf <vrf> ike logging syslog private-authorization
```

default (deprecated)

Default max log level.

```
vrouter running config# vrf <vrf> ike logging syslog private-authorization
vrouter running private-authorization# 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.

asn1 (deprecated)

Low-level encoding/decoding (ASN.1, X.509 etc.).

```
vrouter running config# vrf <vrf> ike logging syslog private-authorization
vrouter running private-authorization# 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 (deprecated)

Configuration management and plugins.

```
vrouter running config# vrf <vrf> ike logging syslog private-authorization
vrouter running private-authorization# 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 (deprecated)

CHILD_SA/IPsec SA processing.

```
vrouter running config# vrf <vrf> ike logging syslog private-authorization
vrouter running private-authorization# 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 (deprecated)

Main daemon setup/cleanup/signal handling.

```
vrouter running config# vrf <vrf> ike logging syslog private-authorization
vrouter running private-authorization# 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 (deprecated)

Packet encoding/decoding encryption/decryption operations.

```
vrouter running config# vrf <vrf> ike logging syslog private-authorization
vrouter running private-authorization# 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 (deprecated)

Libipsec library messages.

```
vrouter running config# vrf <vrf> ike logging syslog private-authorization
vrouter running private-authorization# 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 (deprecated)

IKE_SA/ISAKMP SA processing.

```
vrouter running config# vrf <vrf> ike logging syslog private-authorization
vrouter running private-authorization# 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 (deprecated)

Jobs queuing/processing and thread pool management.

```
vrouter running config# vrf <vrf> ike logging syslog private-authorization
vrouter running private-authorization# 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 (deprecated)

IPsec/Networking kernel interface.

```
vrouter running config# vrf <vrf> ike logging syslog private-authorization
vrouter running private-authorization# 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 (deprecated)

Libstrongwan library messages.

```
vrouter running config# vrf <vrf> ike logging syslog private-authorization
vrouter running private-authorization# 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 (deprecated)

IKE_SA manager, handling synchronization for IKE_SA access.

```
vrouter running config# vrf <vrf> ike logging syslog private-authorization
vrouter running private-authorization# 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 (deprecated)

IKE network communication.

```
vrouter running config# vrf <vrf> ike logging syslog private-authorization
vrouter running private-authorization# 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.

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
```

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.

ike-sync (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-06-18	20q1	IKE HA synchronization is now performed over TCP. Rate limiting parameters are no longer taken into account.	None

IKE state synchronization rate limiting (deprecated).

```
vrouter running config# vrf <vrf> ike ha
vrouter running ha# ike-sync max-rate <uint32> max-burst <uint32>
```

max-rate (deprecated)

IKE state synchronization message maximum rate in pps.

```
max-rate <uint32>
```

max-burst (deprecated)

IKE state synchronization message maximum burst in packets.

```
max-burst <uint32>
```

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.

Default value

```
pre-shared-key
```

remote-auth-method (config only)

Remote IKE authentication method.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template>
vrouter 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.

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.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template>
vrouter 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.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template>
vrouter 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.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template>␣
↪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 (config only)

List of combined-mode (AEAD) algorithms for IKE SAs.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template>␣
↪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 (config only)

List of pseudo-random algorithms for IKE SAs.

```
vrouter running config# vrf <vrf> ike ike-policy-template <ike-policy-template>␣
↪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 (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.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
↪ 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 (config only)

List of combined-mode (AEAD) algorithms for IPsec SAs.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
↪ 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 (config only)

List of Diffie Hellman groups for Perfect Forward Secrecy.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
↪ 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 (config only)

List of Extended Sequence Number modes.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
 ↪ esp-proposal <uint8>
vrouter 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 <string>
```

<string>	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 <string>
vrouter running ah-proposal <string># 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 <string>
vrouter running ah-proposal <string># 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.

```
vrouter running config# vrf <vrf> ike ipsec-policy-template <ipsec-policy-template>
 ↪ ah-proposal <string>
vrouter running ah-proposal <string># esn true|false
```

vpn

List of IKE Virtual Private Networks.

```
vrouter 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-address values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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-ADDRESS values	Description
<domain-name>	The domain-name type represents a DNS domain name. Fully quallified 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 val-ues	Description
<ike-id>	An IPv4 address.
<ike-id>	An IPv6 address.
<ike-id>	The domain-name type represents a DNS domain name. Fully quallified 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-ID-type values	Description
<ike-id>	An IPv4 address.
<ike-id>	An IPv6 address.
<ike-id>	The domain-name type represents a DNS domain name. Fully quallified 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).

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>
```

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.

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.

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 <string>
```

<string>	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 <string>
vrouter running ah-proposal <string># 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 <string>
vrouter running ah-proposal <string># 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 <string>
vrouter running ah-proposal <string># 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-resp (state only)

Sent IKE_AUTH responses.

```
vrouter> show state vrf <vrf> ike counters ike-auth-out-resp
```

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-resp (state only)

Received CREATE_CHILD_SA responses.

```
vrouter> show state vrf <vrf> ike counters create-child-in-resp
```

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-resp (state only)

Sent CREATE_CHILD_SA responses.

```
vrouter> show state vrf <vrf> ike counters create-child-out-resp
```

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
```

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)

IKEv2 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
```

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
```

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
```

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
```

3.2.23 sflow

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-collector> 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 quallified 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 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.

```
vrouter running config# vrf <vrf> sflow
vrouter running sflow# 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
```

3.2.24 snmp

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>	An IPv4 address.
<X:X::X:X>	An IPv6 address.
<host-name>	The domain-name type represents a DNS domain name. Fully quallified 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>	An IPv4 prefix: address and CIDR mask.
<X:X::X:X/M>	An 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>
```

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>
```


<destination> 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 quallified 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
```

FREQUENCY	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
```

FREQUENCY	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
```

FREQUENCY	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
```

3.2.25 routing

global

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
```

deny (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-10-16	20q3	The configuration result of mixed deny/permit nodes was not deterministic, and is replaced by an ordered list	/vrouter-routing:routing/ipv4-access-list-config/ipv4-access-list/seq/deny

IPv4 access list deny rules.

```
vrouter running config# routing ipv4-access-list <string>
vrouter running ipv4-access-list <string># deny <deny> exact-match true|false
```


<deny> values	Description
<A.B.C.D/M>	An IPv4 prefix: address and CIDR mask.
any	Any prefix.

exact-match (deprecated)

Enable or disable exact match of the prefixes.

```
exact-match true|false
```

permit (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-10-16	20q3	The configuration result of mixed deny/permit nodes was not deterministic, and is replaced by an ordered list	/vrouter-routing:routing/ipv4-access-list-config/ipv4-access-list/seq/permit

IPv4 access list permit rules.

```
vrouter running config# routing ipv4-access-list <string>
vrouter running ipv4-access-list <string># permit <permit> exact-match true|false
```

<permit> values	Description
<A.B.C.D/M>	An IPv4 prefix: address and CIDR mask.
any	Any prefix.

exact-match (deprecated)

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
```

deny (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-10-16	20q3	The configuration result of mixed deny/permit nodes was not deterministic, and is replaced by an ordered list	/vrouter-routing:routing/ipv6-access-list-config/ipv6-access-list/seq/deny

IPv6 access list deny rules.

```
vrouter running config# routing ipv6-access-list <string>
vrouter running ipv6-access-list <string># deny <deny> exact-match true|false
```

<deny> values	Description
<X:X::X:X/M>	An IPv6 prefix: address and CIDR mask.
any	Any prefix.

exact-match (deprecated)

Enable or disable exact match of the prefixes.

```
exact-match true|false
```

permit (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-10-16	20q3	The configuration result of mixed deny/permit nodes was not deterministic, and is replaced by an ordered list	/vrouter-routing:routing/ipv6-access-list-config/ipv6-access-list/seq/permit

IPv6 access list permit rules.

```
vrouter running config# routing ipv6-access-list <string>
vrouter running ipv6-access-list <string># permit <permit> exact-match true|false
```

<permit> values	Description
<X:X::X:X/M>	An IPv6 prefix: address and CIDR mask.
any	Any prefix.

exact-match (deprecated)

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

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.

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.

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.

```
vrouter running config# routing logging bgp updates
vrouter 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.

```
vrouter running config# routing logging bgp vpn
```

label

Log VPN label.

```
vrouter running config# routing logging bgp vpn
vrouter 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

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

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

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

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>
```

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>
```

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
```

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.
substract-metric	Subtract metric.
substract-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
```

as-number (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-01-29	19q3	The same AS number can be prepended several times. It can not be achieved with a leaf-list type, a list type must be used.	/vrouter-routing:routing/route-map/name/seq/set/as-path/prepend/asn

AS numbers to prepend to the as-path.

```
vrouter running config# routing route-map <string> seq <uint16> set as-path prepend
vrouter running prepend# as-number <uint32>
```

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.

```
vrouter running config# routing route-map <string> seq <uint16> set as-path␣
↪prepend asn <uint8>
```

<uint8>	Priority of the AS number. High number means lower priority.

<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 address.

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
```

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.
additive	Additive.
<string>	Community attribute.

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 SOO
```

SOO values	Description
<string>	Extended communities or route-target attribute.
<string>	Extended communities or route-target attribute.
<string>	Extended communities or route-target attribute.

bgp

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

ipv4-community-list (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-04-05	20q1	Community lists and IP version are independent of each other. The 'ipv4-' prefix was misleading. There is also only one deny and permit configurable list per community-list whereas there should be several. The new 'community-list' configuration option fixes this.	/vrouter-routing:routing/bgp/commu list

Add a community list entry.

```
vrouter running config# routing bgp ipv4-community-list <ipv4-community-list>
```

<ipv4-community-list> values	Description
<uint8>	List name.
<string>	List name.

deny (deprecated)

Specify communities to reject.

```
vrouter running config# routing bgp ipv4-community-list <ipv4-community-list>
vrouter running ipv4-community-list <ipv4-community-list># deny DENY
```

DENY values	Description
local-AS	Local AS.
no-advertise	Do not advertise.
no-export	Do not export.
internet	Internet.
additive	Additive.
<string>	Community attribute.

permit (deprecated)

Specify communities to accept.

```
vrouter running config# routing bgp ipv4-community-list <ipv4-community-list>
vrouter running ipv4-community-list <ipv4-community-list># permit PERMIT
```

PERMIT values	Description
local-AS	Local AS.
no-advertise	Do not advertise.
no-export	Do not export.
internet	Internet.
additive	Additive.
<string>	Community attribute.

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.
additive	Additive.
<string>	Community attribute.

ipv4-extcommunity-list (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-04-05	20q1	Extcommunity lists and IP version are independent of each other. The 'ipv4-' prefix was misleading. There is also only one deny and permit configurable list per extcommunity whereas there should be several. The new 'extcommunity-list' configuration option fixes this.	/vrouter-routing:routing/bgp/extcomm list

Add an extended community list entry.

```
vrouter running config# routing bgp ipv4-extcommunity-list <ipv4-extcommunity-list>
```

<ipv4-extcommunity-list> values	Description
<uint8>	List name.
<string>	List name.

deny (deprecated)

Specify extended communities to reject.

```
vrouter running config# routing bgp ipv4-extcommunity-list <ipv4-extcommunity-list>
↪ deny
```

rt (deprecated)

Extended community route target to reject.

```
vrouter running config# routing bgp ipv4-extcommunity-list <ipv4-extcommunity-list>
↪ deny
vrouter running deny# 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 (deprecated)

Extended community site of origin to reject.

```
vrouter running config# routing bgp ipv4-extcommunity-list <ipv4-extcommunity-list>
↪ deny
vrouter running deny# soo SOO
```

SOO values	Description
<string>	Extended communities or route-target attribute.
<string>	Extended communities or route-target attribute.
<string>	Extended communities or route-target attribute.

permit (deprecated)

Specify extended communities to accept.

```
vrouter running config# routing bgp ipv4-extcommunity-list <ipv4-extcommunity-list>
↪ permit
```

rt (deprecated)

Extended community route target to accept.

```
vrouter running config# routing bgp ipv4-extcommunity-list <ipv4-extcommunity-list>
↪ permit
vrouter running permit# 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 (deprecated)

Extended community site of origin to accept.

```
vrouter running config# routing bgp ipv4-extcommunity-list <ipv4-extcommunity-list>
↪ permit
vrouter running permit# soo SOO
```

SOO 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

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 SOO
```

<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 SOO
```

SOO values	Description
<string>	Extended communities or route-target attribute.
<string>	Extended communities or route-target attribute.
<string>	Extended communities or route-target attribute.

ipv4-as-path-access-list (deprecated)

Deprecated since	Obsolete in release	Description	Replacement
2019-04-05	20q1	AS path access-list and IP version are independent of each other. The 'ipv4-' prefix was misleading. There is also only one deny and permit configurable list per AS path whereas there should be several. The new 'as-path-access-list' configuration option fixes this.	/vrouter-routing:routing/bgp/as-path-access-list

BGP autonomous system path filter.

```
vrouter running config# routing bgp ipv4-as-path-access-list <string>
```

<string>	Access list name.

deny (deprecated)

Specify a regular expression that match AS paths to reject.

```
vrouter running config# routing bgp ipv4-as-path-access-list <string>
vrouter running ipv4-as-path-access-list <string># deny <string>
```

permit (deprecated)

Specify a regular expression that match AS paths to accept.

```
vrouter running config# routing bgp ipv4-as-path-access-list <string>
vrouter running ipv4-as-path-access-list <string># permit <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 table <uint32> 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.

table

Table to configure.

```
table <uint32>
```

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 table <uint32> 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.

table

Table to configure.

```
table <uint32>
```

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 ospf

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

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

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.

```
vrouter running config# vrf <vrf> routing interface <interface> ip rip
vrouter 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 recpetion/transmission version.

```
vrouter running config# vrf <vrf> routing interface <interface> ip rip version
```

receive

Advertisement reception - Version control.

```
vrouter running config# vrf <vrf> routing interface <interface> ip rip version
vrouter 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

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
```

bgp

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.

```
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
```

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.

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>
```

Default value

```
15
```

reuse

Value to start reusing a route.

```
vrouter running config# vrf <vrf> routing bgp dampening
vrouter running dampening# reuse <uint16>
```

Default value

```
750
```

suppress

Value to start suppressing a route.

```
vrouter running config# vrf <vrf> routing bgp dampening
vrouter running dampening# suppress <uint16>
```

Default value

```
2000
```


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>
```

Default value

```
60
```

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

```
10
```

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 addres 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
```

Default value

```
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.

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
```

med (state only)

Multi Exit Discriminator.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network
↪<network> route <string> med
```

origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network
↪<network> route <string> origin
```

packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-unicast network
↪<network> route <string> packet-length
```

nexthop (state only)

Route nexthop.

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
```

<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
```

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
```

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
```

Default value

```
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.

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
```

med (state only)

Multi Exit Discriminator.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network
↪<network> route <string> med
```

origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network
↪<network> route <string> origin
```

packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-multicast network
↪<network> route <string> packet-length
```

nexthop (state only)

Route nexthop.

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
```

<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
```

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
```

ipv4-flowspec

Configure IPv4 Flowspec address family.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-flowspec
```

enabled

Enable or disable IPv4 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
```

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
```

med (state only)

Multi Exit Discriminator.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route
↪<uint32> med
```

origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route
↪<uint32> origin
```

packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route
↪<uint32> packet-length
```

nexthop (state only)

Route nexthop.

used (state only)

Nexthop used.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-flowspec route
↪<uint32> nexthop <nexthop> used
```

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> 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-prefix>	An IPv4 prefix: address and CIDR mask.

label

VPN NLRI label.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-vpn route-
↪distinguisher <rd> <ip-prefix>
vrouter running route-distinguisher <rd> <ip-prefix># label <uint32>
```

route-map

Route map name.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv4-vpn route-
↪distinguisher <rd> <ip-prefix>
vrouter running route-distinguisher <rd> <ip-prefix># route-map ROUTE-MAP
```

ROUTE-MAP	Route map name.

route (state only)

Route operational state.

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
```

med (state only)

Multi Exit Discriminator.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-
↪distinguisher <rd> <ip-prefix> route <string> med
```

origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv4-vpn route-
↪distinguisher <rd> <ip-prefix> route <string> origin
```

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
```

nexthop (state only)

Route nexthop.

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.

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
```

med (state only)

Multi Exit Discriminator.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network
↪<network> route <string> med
```

origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network
↪<network> route <string> origin
```

packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-unicast network
↪<network> route <string> packet-length
```

nexthop (state only)

Route nexthop.

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> summary-only␣
↪true|false
```

<aggregate-address>	An IPv6 prefix: address and CIDR mask.

summary-only

If true, filter more specific routes from updates.

```
summary-only true|false
```

Default value

```
false
```

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.

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.

route-target

Route target list.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-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 ipv6-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 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-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 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
```

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.

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
```

med (state only)

Multi Exit Discriminator.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network
↪<network> route <string> med
```

origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network
↪<network> route <string> origin
```

packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-multicast network
↪<network> route <string> packet-length
```

nexthop (state only)

Route nexthop.

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
```

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
```

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> 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-prefix>	An IPv6 prefix: address and CIDR mask.

label

VPN NLRI label.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-vpn route-
↪distinguisher <rd> <ip-prefix>
vrouter running route-distinguisher <rd> <ip-prefix># label <uint32>
```

route-map

Route map name.

```
vrouter running config# vrf <vrf> routing bgp address-family ipv6-vpn route-
↪distinguisher <rd> <ip-prefix>
vrouter running route-distinguisher <rd> <ip-prefix># route-map ROUTE-MAP
```

ROUTE-MAP	Route map name.

route (state only)

Route operational state.

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
```

med (state only)

Multi Exit Discriminator.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-
↪distinguisher <rd> <ip-prefix> route <string> med
```

origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-vpn route-
↪distinguisher <rd> <ip-prefix> route <string> origin
```

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
```

nexthop (state only)

Route nexthop.

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
```

ipv6-flowspec (state only)

Configure IPv6 Flowspec address family.

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
```

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
```

med (state only)

Multi Exit Discriminator.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route
↪<uint32> med
```

origin (state only)

Route origin.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route
↪<uint32> origin
```

packet-length (state only)

Packet length.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route
↪<uint32> packet-length
```

nexthop (state only)

Route nexthop.

used (state only)

Nexthop used.

```
vrouter> show state vrf <vrf> routing bgp address-family ipv6-flowspec route
↪<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. If the tracked address is reachable, the neighbor is considered as valid, else it is disabled.

```
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.

```
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.

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 miliseconds.

```
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-rount-trip-time (state only)

Estimated round trip time in miliseconds.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> estimated-rount-
↪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.

```
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.

```
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
```

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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv4-unicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv4-unicast
vrouter running ipv4-unicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

ATTRIBUTE-UNCHANGED values	Description
as-path	As-path attribute.
med	Med attribute.
nexthop	Nexthop attribute.

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
```

route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv4-unicast
vrouter running ipv4-unicast# route-reflector-client 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
```


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
```

distribute-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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv4-unicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv4-unicast
vrouter 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
```

send-community

Send Community attribute to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv4-multicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv4-multicast
vrouter running ipv4-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 ipv4-multicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv4-multicast
vrouter running ipv4-multicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

ATTRIBUTE-UNCHANGED values	Description
as-path	As-path attribute.
med	Med attribute.
nexthop	Nexthop attribute.

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-multicast
vrouter 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
```

route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv4-multicast
vrouter running ipv4-multicast# route-reflector-client 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
```

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
```


distribute-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# 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-multicast maximum-prefix
```

maximum (mandatory)

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv4-multicast 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-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
```


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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

ATTRIBUTE-UNCHANGED values	Description
as-path	As-path attribute.
med	Med attribute.
nexthop	Nexthop attribute.

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-labeled-unicast
vrouter 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
```


route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# route-reflector-client 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
```

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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> 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-group <string> 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-group <string> 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-group <string> 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.

```
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-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
```

route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
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
```

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
```


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
```

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-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
```

route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
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
```

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
```

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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv4-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 ipv4-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 ipv4-vpn
vrouter running ipv4-vpn# distribute-list <distribute-list>
```

<distribute-list> values	Description
in	Filter incoming updates.
out	Filter outgoing updates.

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>
```

<filter-list> values	Description
in	Filter incoming updates.
out	Filter outgoing updates.

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> values	Description
in	Filter incoming updates.
out	Filter outgoing updates.

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> values	Description
in	Apply map to incoming routes.
out	Apply map to outbound routes.

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
```


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-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> 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-group <string> address-
↪family ipv6-unicast
vrouter running ipv6-unicast# soft-reconfiguration-inbound true|false
```

Default value

```
false
```

route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv6-unicast
vrouter running ipv6-unicast# route-reflector-client 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
```

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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv6-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-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 ipv6-unicast
vrouter 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
```

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-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
```

route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
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
```

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
```

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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv6-multicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv6-multicast maximum-prefix
```

maximum (mandatory)

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv6-multicast 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-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
```

send-community

Send Community attribute to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv6-labeled-unicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv6-labeled-unicast
vrouter running ipv6-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 ipv6-labeled-unicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv6-labeled-unicast
vrouter running ipv6-labeled-unicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

ATTRIBUTE-UNCHANGED values	Description
as-path	As-path attribute.
med	Med attribute.
nexthop	Nexthop attribute.

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-labeled-unicast
vrouter 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
```


route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv6-labeled-unicast
vrouter running ipv6-labeled-unicast# route-reflector-client 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
```

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
```

distribute-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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> 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-group <string> 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-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-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
```


route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
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
```

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
```


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.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> address-
↪family ipv6-vpn
vrouter running ipv6-vpn# as-override true|false
```

Default value

```
false
```

send-community

Send Community attribute to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor-group <string> 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-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-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
```

route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
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
```

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
```

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>
```

<distribute-list> values	Description
in	Filter incoming updates.
out	Filter outgoing updates.

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>
```

<filter-list> values	Description
in	Filter incoming updates.
out	Filter outgoing updates.

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> values	Description
in	Filter incoming updates.
out	Filter outgoing updates.

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> values	Description
in	Apply map to incoming routes.
out	Apply map to outbound routes.

connections (state only)

Established/dropped connections statistics.

established (state only)

Number of established connections.

```
vrouter> show state vrf <vrf> routing bgp neighbor-group <string> connections␣
↪established
```

dropped (state only)

Number of dropped connections.

```
vrouter> 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-nexthop	Advertise extended nexthop 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. If the tracked address is reachable, the neighbor is considered as valid, else it is disabled.

```
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.

```
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.

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 miliseconds.

```
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-rount-trip-time (state only)

Estimated round trip time in miliseconds.

```
vrouter> show state vrf <vrf> routing bgp neighbor <neighbor> estimated-rount-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.

```
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.

```
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
```

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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-unicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-unicast
vrouter running ipv4-unicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

ATTRIBUTE-UNCHANGED values	Description
as-path	As-path attribute.
med	Med attribute.
nexthop	Nexthop attribute.

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
```

route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-unicast
vrouter running ipv4-unicast# route-reflector-client 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
```


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
```

distribute-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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> 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 <neighbor> address-family␣
↪ipv4-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-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-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␣
↪ipv4-unicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-unicast
vrouter 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
```

send-community

Send Community attribute to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-multicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-multicast
vrouter running ipv4-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␣
↪ipv4-multicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-multicast
vrouter running ipv4-multicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

ATTRIBUTE-UNCHANGED values	Description
as-path	As-path attribute.
med	Med attribute.
nexthop	Nexthop attribute.

route-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-multicast
vrouter 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
```

route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-multicast
vrouter running ipv4-multicast# route-reflector-client 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
```

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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-multicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-multicast maximum-prefix
```

maximum (mandatory)

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-multicast 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-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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> 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 <neighbor> address-family␣
↪ipv4-multicast as-outbound-update
```

action

Action to apply for ASNs in outbound updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> 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 <neighbor> 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 <neighbor> 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 <neighbor> 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 <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
```

send-community

Send Community attribute to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> 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 <neighbor> 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 <neighbor> 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

`ATTRIBUTE-UNCHANGED` values	Description
as-path	As-path attribute.
med	Med attribute.
nexthop	Nexthop attribute.

route-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-labeled-unicast
vrouter 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
```

route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-labeled-unicast
vrouter running ipv4-labeled-unicast# route-reflector-client 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
```

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
```

distribute-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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> 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 <neighbor> 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 <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-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
```


route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
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
```

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
```


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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> 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 <neighbor> 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 <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
```

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-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
```

route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
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
```

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
```

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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-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 <neighbor> address-family␣
↪ipv4-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 <neighbor> address-family␣
↪ipv4-vpn
vrouter running ipv4-vpn# distribute-list <distribute-list>
```

<distribute-list> values	Description
in	Filter incoming updates.
out	Filter outgoing updates.

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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-vpn
vrouter running ipv4-vpn# filter-list <filter-list>
```

<filter-list> values	Description
in	Filter incoming updates.
out	Filter outgoing updates.

prefix-list

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv4-vpn
vrouter running ipv4-vpn# prefix-list <prefix-list>
```

<prefix-list> values	Description
in	Filter incoming updates.
out	Filter outgoing updates.

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> values	Description
in	Apply map to incoming routes.
out	Apply map to outbound routes.

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
```

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-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
```


route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
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
```

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
```

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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv6-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-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 <neighbor> address-family␣
↪ipv6-unicast
vrouter 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
```

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-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
```

route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
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
```

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
```

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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv6-multicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv6-multicast maximum-prefix
```

maximum (mandatory)

Maximum number of prefix limit.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv6-multicast 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-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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> 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 <neighbor> address-family␣
↪ipv6-multicast as-outbound-update
```

action

Action to apply for ASNs in outbound updates.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> 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 <neighbor> 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 <neighbor> 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 <neighbor> 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 <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
```

send-community

Send Community attribute to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv6-labeled-unicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv6-labeled-unicast
vrouter running ipv6-labeled-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-labeled-unicast
vrouter 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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv6-labeled-unicast
vrouter running ipv6-labeled-unicast# attribute-unchanged ATTRIBUTE-UNCHANGED
```

`ATTRIBUTE-UNCHANGED` values	Description
as-path	As-path attribute.
med	Med attribute.
nexthop	Nexthop attribute.

route-server-client

If true, configure a neighbor as Route Server client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv6-labeled-unicast
vrouter 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
```

route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv6-labeled-unicast
vrouter running ipv6-labeled-unicast# route-reflector-client 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
```

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
```

distribute-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-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
```


route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
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
```

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
```


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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> 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 <neighbor> 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 <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
```

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-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
```

route-reflector-client

If true, configure a neighbor as Route Reflector client.

```
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
```

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
```

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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> 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 <neighbor> 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 <neighbor> address-family␣
↪ipv6-vpn
vrouter running ipv6-vpn# distribute-list <distribute-list>
```

<distribute-list> values	Description
in	Filter incoming updates.
out	Filter outgoing updates.

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.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv6-vpn
vrouter running ipv6-vpn# filter-list <filter-list>
```

<filter-list> values	Description
in	Filter incoming updates.
out	Filter outgoing updates.

prefix-list

Filter updates to/from this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv6-vpn
vrouter running ipv6-vpn# prefix-list <prefix-list>
```

<prefix-list> values	Description
in	Filter incoming updates.
out	Filter outgoing updates.

route-map

Apply route map to this neighbor.

```
vrouter running config# vrf <vrf> routing bgp neighbor <neighbor> address-family␣
↪ipv6-vpn
vrouter running ipv6-vpn# route-map <route-map>
```

<route-map> values	Description
in	Apply map to incoming routes.
out	Apply map to outbound routes.

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
```


ldp

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.

ospf

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>
```

Default value

```
1
```


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>
```

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.

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>
```

<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>
```

rip

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

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

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
```

distribute-list

Filter networks in routing updates.

```
vrouter running config# vrf <vrf> routing ripng
vrouter running ripng# 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 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

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
```

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
```

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
```

3.2.26 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 ommitted, 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>
```

3.2.27 fast-path

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 <leafref>
```

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.
dpdk-pmd	DPDK crypto PMD.
octeontxcpt	Marvell Octeon TX.

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	A comma-separated list of cores or core ranges. Example: '1,4-7,10-12'.

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>
```

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>
```

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

Maximum number of IPv4 and IPv6 /8 table entries.

```
vrouter running config# system fast-path limits
vrouter running limits# ip-max-8-table <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>
```

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
```

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
```

3.2.28 logging

Global Settings

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

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>	An IPv4 address.
<X:X::X:X>	An IPv6 address.
<host-name>	The domain-name type represents a DNS domain name. Fully quallified 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>
```

4. Troubleshooting

This guide references common configuration issues one may encounter when using Turbo IPsec, 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.

4.2 Typical issues

4.2.1 Startup Issues

Turbo IPsec 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 IPsec versions. Some shells (bash for example) keep a cache of the executable paths. After upgrading Turbo IPsec, if some commands are not found, you may need to start a new shell.

Hugepages fragmentation

Symptoms

- One of the following messages appears on he 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 fpn_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 *meminfo* 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 he 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 he 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 he 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 IPsec 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 IPsec hangs when starting with i40e devices

Symptoms

- Starting Turbo IPsec 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]  0000000000000086 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] [<ffffffffc01d0230>] 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
[  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 License not found

Symptoms

- Trying to load a license fails.
- The following messages can be found in the logs:

```
LICENSE: License hostid determination failed
LICENSE: License checkout for turbo-router v00.08.09
LICENSE: License initialization failed:
#011Can't read license data (-102)No such file or directory (errno: 2)

LICENSE: failed to acquire valid license, launching time bomb
        error: unsupported configuration: host doesn't support␣
↪passthrough of host PCI devices
```

Hints

- Ensure your license file is installed and can be read: `chmod 0644 /path/to/your/license/file.lic`

4.2.3 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.
- `ethtool -S <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` man-page 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 IPsec, 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 IPsec 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
```

4.2.4 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
```

4.2.5 OpenStack

This section gathers issues that happen with Turbo IPsec 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 State | Networks                     |
+--------------------------------------+------+--------+------------+-----
↪--------+------------------------------+
| 52ad953d-19dd-47a9-b03d-dfe565e655e1 | vm3  | ERROR  | -          |␣
↪NOSTATE     |                              |
| 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 `ERROR`s and `TRACE`s 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 `ERROR`s and `TRACE`s 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.

4.3 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
fpn0-vr0 port:254
ntfp4-vr0 port:0
ntfp1-vr0 port:1
ntfp2-vr0 port:2
ntfp3-vr0 port:3
==== global stats:
```

(continues on next page)

(continued from previous page)

```
==== 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=0x00007faee19be9f9 thread_id=26774
  CPU #2: pc=0xffffffff8104f596 (halted) thread_id=26775
  CPU #3: pc=0x0000000000530233 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 `vcpupin` to pin virtual CPUs to physical ones. e.g.:

```
<vcpu cpuset='7-8,27-28'>4</vcpu>
<cputune>
```

(continues on next page)

(continued from previous page)

```
  <vcpupin vcpu="0" cpuset="7"/>
  <vcpupin vcpu="1" cpuset="8"/>
  <vcpupin vcpu="2" cpuset="27"/>
  <vcpupin 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       VIRT   RES   SHR S CPU% MEM%   TIME+  NLWP Command
26770 mazon     7032M 4067M  7092 S 200. 25.5  2h19:55    7 |- qemu-system-x86_64 -
↪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 -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 -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 -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 -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 -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 -daemonize --enable-kvm -m 6G -cpu host -smp sockets=1,cores=4,threads=1 ...
```

You may even change CPU affinity by typing `a` 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 ethtool

The `ethtool` program 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 `ethtool -S <port>`:

```
# ethtool -S eth1

NIC statistics:
     rx_good_packets: 261064663
     tx_good_packets: 256512062
```

(continues on next page)

(continued from previous page)

```
     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.

`ethtool` can be used to check whether offload is enabled, using the following:

```
# ethtool -k <port>
Features for <port>:
rx-checksumming: off [fixed]
tx-checksumming: off
        tx-checksum-ipv4: off [fixed]
        tx-checksum-ip-generic: off [fixed]
        tx-checksum-ipv6: off [fixed]
        tx-checksum-fcoe-crc: off [fixed]
        tx-checksum-sctp: off [fixed]
scatter-gather: off
        tx-scatter-gather: off [fixed]
        tx-scatter-gather-fraglist: off [fixed]
```

(continues on next page)

(continued from previous page)

```
tcp-segmentation-offload: off
        tx-tcp-segmentation: off [fixed]
        tx-tcp-ecn-segmentation: off [fixed]
        tx-tcp6-segmentation: off [fixed]
udp-fragmentation-offload: off [fixed]
generic-segmentation-offload: off [requested on]
generic-receive-offload: on
large-receive-offload: off [fixed]
rx-vlan-offload: off [fixed]
tx-vlan-offload: off [fixed]
ntuple-filters: off [fixed]
receive-hashing: off [fixed]
highdma: off [fixed]
rx-vlan-filter: off [fixed]
vlan-challenged: off [fixed]
tx-lockless: off [fixed]
netns-local: off [fixed]
tx-gso-robust: off [fixed]
tx-fcoe-segmentation: off [fixed]
tx-gre-segmentation: off [fixed]
tx-ipip-segmentation: off [fixed]
tx-sit-segmentation: off [fixed]
tx-udp_tnl-segmentation: off [fixed]
tx-mpls-segmentation: off [fixed]
fcoe-mtu: off [fixed]
tx-nocache-copy: off
loopback: off [fixed]
rx-fcs: off [fixed]
rx-all: off [fixed]
tx-vlan-stag-hw-insert: off [fixed]
rx-vlan-stag-hw-parse: off [fixed]
rx-vlan-stag-filter: off [fixed]
busy-poll: off [fixed]
```

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:

```
# ethtool -K 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.

Note: Refer to the `ethtool` manpage for specific options.

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]
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 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 PIIX3 ISA [Natoma/Triton II]
        Subsystem: Red Hat, Inc Qemu virtual machine
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 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 PIIX4 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
```

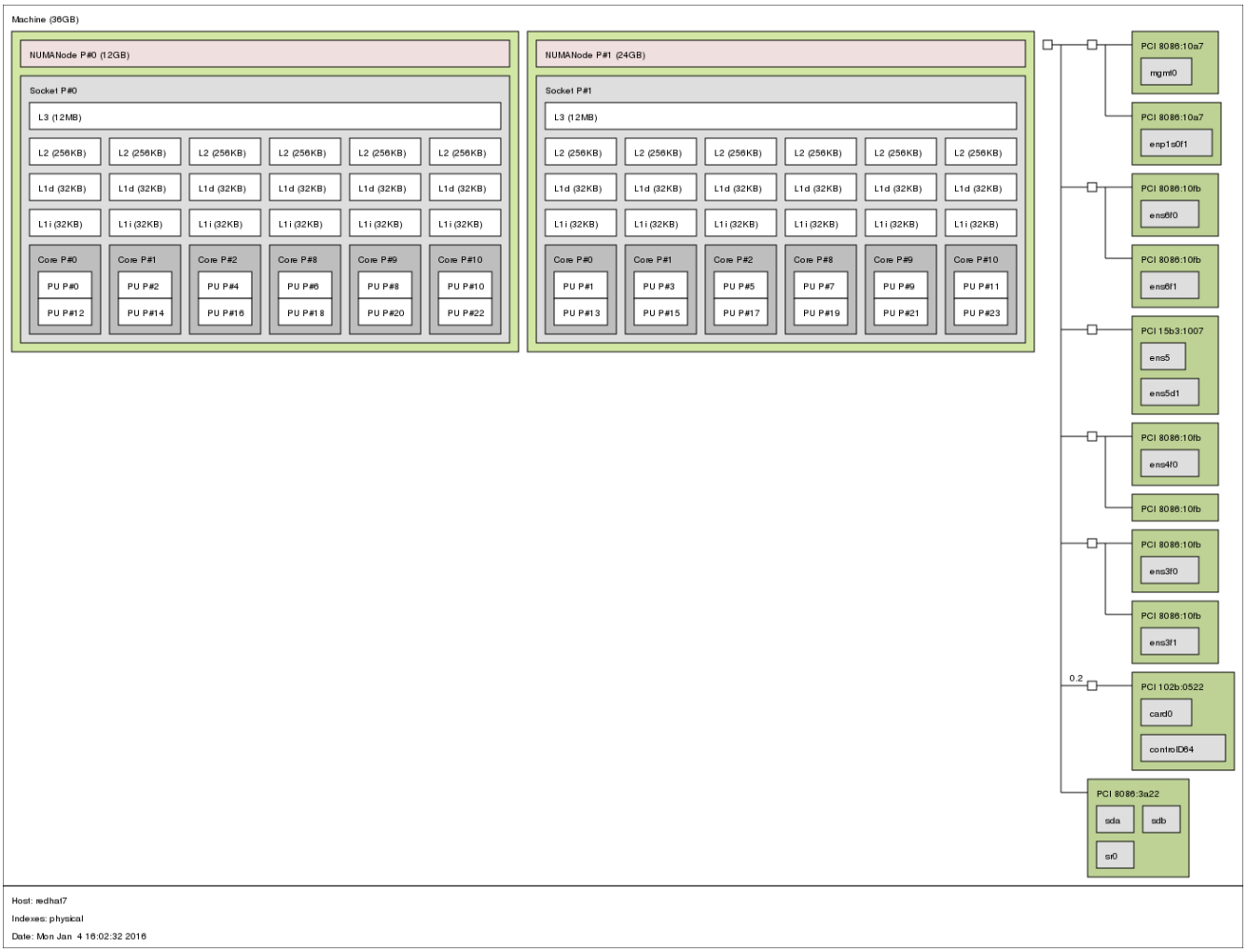
Note: Refer to the `lspci` manpage for specific options.

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
Cached:           626528 kB
```

(continues on next page)

(continued from previous page)

```
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).
For details regarding the `HugePages` fields, look there (http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Docume

---

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]: libfpn_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
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
```

(continues on next page)

(continued from previous page)

```
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
```

(continues on next page)

(continued from previous page)

```
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 --
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
```

(continues on next page)

(continued from previous page)

```
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
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.
```

(continues on next page)

(continued from previous page)

```
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
|   |-- 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>
```

(continues on next page)

(continued from previous page)

```
  </memoryBacking>
  <numatune>
```

Or you can look at a currently running Nova instance console logs:

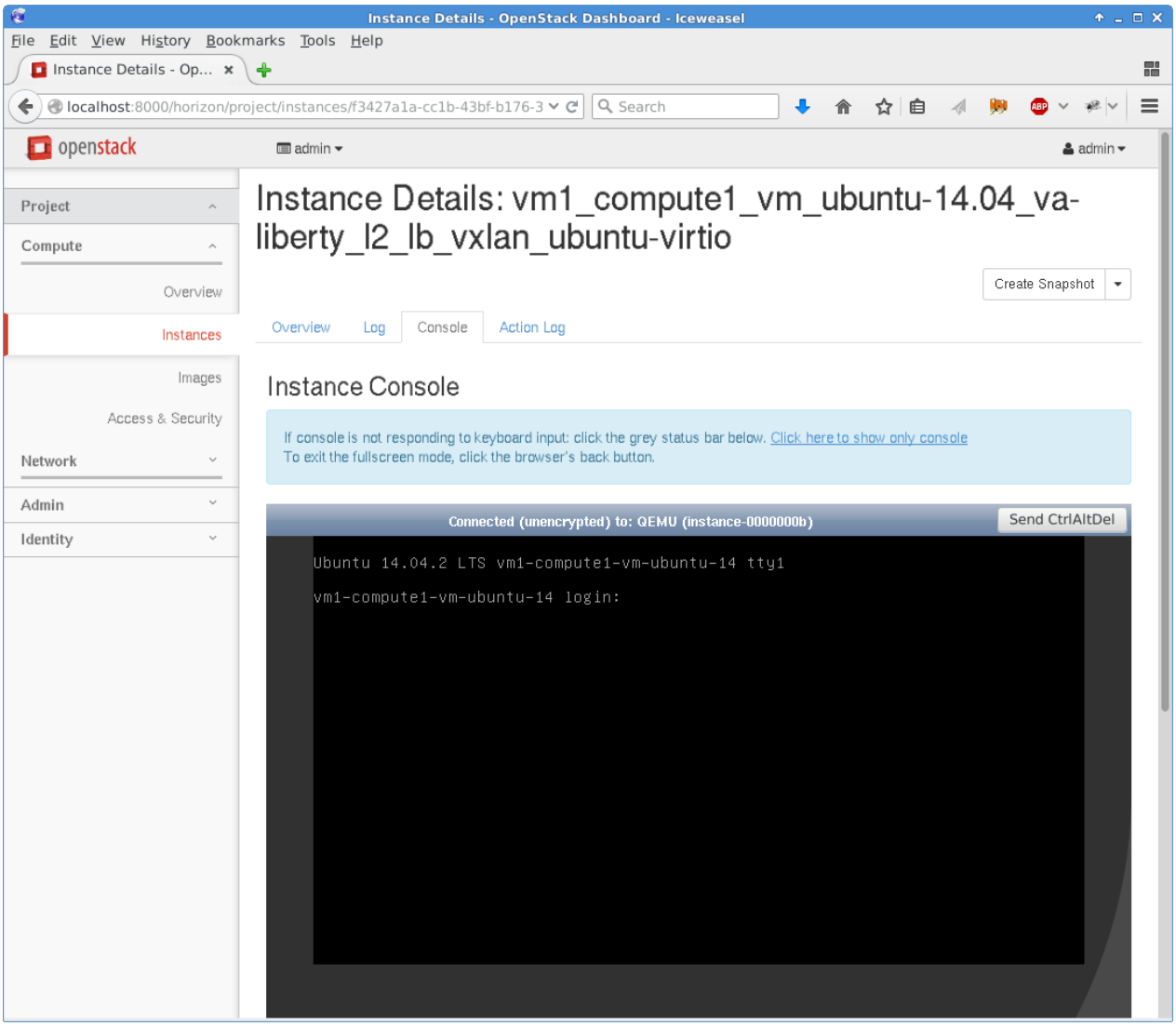
```
# tail /var/lib/nova/instances/54fff47b-fa5e-4401-8309-e2da66c01d66/console.log
ec2: #############################################################
-----BEGIN SSH HOST KEY KEYS-----
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYT[...]Osfj0fFcxJvE2Roc= root@compute1-vm
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQ[...]JUYLnAaNV8oNz1AId 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 obviously).

  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
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
```

(continues on next page)

(continued from previous page)

```
epoll_wait(4,
^C
Process 5350 detached
 <detached ...>
```

Note: Refer to the `strace` manpage for specific options.