# 7450 Ethernet Service Switch
# 7750 Service Router
# 7950 Extensible Routing System
# Virtualized Service Router

Release 24.7.R1

## Router Configuration Guide

3HE 20112 AAAB TQZZA 01
Edition: 01
July 2024

Nokia is committed to diversity and inclusion. We are continuously reviewing our customer documentation and consulting with standards bodies to ensure that terminology is inclusive and aligned with the industry. Our future customer documentation will be updated accordingly.

---

# Table of contents

# 1 Getting started

## 1.1 About this guide

This guide describes logical IP routing interfaces, virtual routers, IP and MAC-based filtering, and cflowd support and presents configuration and implementation examples.

This guide is organized into functional chapters and provides concepts and descriptions of the implementation flow, as well as Command Line Interface (CLI) syntax and command usage.

**Note:** Unless otherwise indicated, CLI commands, contexts, and configuration examples in this guide apply for both the MD-CLI and the classic CLI.

The topics and commands described in this document apply to the:

- 7450 ESS
- 7750 SR
- 7950 XRS
- Virtualized Service Router

For a list of unsupported features by platform and chassis, see *SR OS R24.x.Rx Software Release Notes*, part number 3HE 20152 000*x* TQZZA.

Command outputs shown in this guide are examples only; actual displays may differ depending on supported functionality and user configuration.

**Note:**
The SR OS CLI trees and command descriptions can be found in the following guides:

- *7450 ESS, 7750 SR, 7950 XRS, and VSR Classic CLI Command Reference Guide*
- *7450 ESS, 7750 SR, 7950 XRS, and VSR Clear, Monitor, Show, and Tools CLI Command Reference Guide* (for both MD-CLI and Classic CLI)
- *7450 ESS, 7750 SR, 7950 XRS, and VSR MD-CLI Command Reference Guide*

**Note:**
This guide generically covers Release 24.*x*.R*x* content and may contain some content that may be released in later maintenance loads. See *SR OS R24.x.Rx Software Release Notes*, part number 3HE 20152 000*x* TQZZA for information about features supported in each load of the Release 24.*x*.R*x* software.

## 1.2 Conventions

This section describes the general conventions used in this guide.

### 1.2.1 Precautionary and information messages

The following information symbols are used in the documentation.

**DANGER:** Danger warns that the described activity or situation may result in serious personal injury or death. An electric shock hazard could exist. Before you begin work on this equipment, be aware of hazards involving electrical circuitry, be familiar with networking environments, and implement accident prevention procedures.

**WARNING:** Warning indicates that the described activity or situation may, or will, cause equipment damage, serious performance problems, or loss of data.

**Caution:** Caution indicates that the described activity or situation may reduce your component or system performance.

**Note:** Note provides additional operational information.

**Tip:** Tip provides suggestions for use or best practices.

### 1.2.2 Options or substeps in procedures and sequential workflows

Options in a procedure or a sequential workflow are indicated by a bulleted list. In the following example, at step 1, the user must perform the described action. At step 2, the user must perform one of the listed options to complete the step.

**Example: Options in a procedure**

1. User must perform this step.
2. This step offers three options. User must perform one option to complete this step.
   - This is one option.
   - This is another option.
   - This is yet another option.

Substeps in a procedure or a sequential workflow are indicated by letters. In the following example, at step 1, the user must perform the described action. At step 2, the user must perform two substeps (a. and b.) to complete the step.

**Example: Substeps in a procedure**

1. User must perform this step.
2. User must perform all substeps to complete this action.
   a. This is one substep.
   b. This is another substep.

# 2 IP router configuration

## 2.1 Configuring IP router command options

To provision services on a Nokia router, logical IP routing interfaces must be configured to associate attributes such as an IP address, port, or the system with the IP interface.

A special type of IP interface is the system interface. A system interface must have an IP address with a 32-bit subnet mask. The system interface is used as the router identifier by higher-level protocols such as OSPF and BGP, unless overwritten by an explicit router ID.

The following router features can be configured:

- Interfaces
- Autonomous systems
- Confederations
- Proxy ARP

See the *7450 ESS, 7750 SR, and VSR Triple Play Service Delivery Architecture Guide* for information about DHCP and support as well as configuration examples for the 7750 SR and 7450 ESS.

### 2.1.1 Interfaces

Nokia routers use different types of interfaces for various functions. Interfaces must be configured with information such as the interface type (network and system) and address. A port is not associated with a system interface. An interface can be associated with the system (loopback address).

#### 2.1.1.1 Network interface

A network interface (a logical IP routing interface) can be configured on one of the following entities:

- physical or logical port
- a SONET/SDH channel for the 7750 SR or 7450 ESS

#### 2.1.1.2 Network domains

To determine which network ports (and, therefore, which network complexes) are eligible to transport traffic of individual SDPs, network-domain is provided. Network-domain information is then used for the sap-ingress queue allocation algorithm applied to VPLS SAPs. This algorithm is optimized in so that no sap-ingress queues are allocated if the specified port does not belong to the network-domain used in the specified VPLS. Also, sap-ingress queues are not allocated toward network ports (regardless of the network-domain membership) if the specified VPLS does not contain any SDPs.

SAP-ingress queue allocation considers the following:

- SHG membership of individual SDPs
- network-domain definition under SDP to restrict the topology in which the specified SDP can be set-up

The implementation supports four network-domains within any VPLS.

Network-domain configuration at the SDP level is ignored when the SDP is used for Epipe or Ipipe bindings.

Network-domain configurations are irrelevant for Layer 3 services (Layer 3 VPN and IES services). Network-domain configurations can be defined in the base routing context and associated only with network interfaces in this context. Network domains are not applicable to loopback and system interfaces.

The network-domain information is only used for ingress VPLS sap queue-allocation. It is not considered by routing during SDP setup. Therefore, if the specified SDP is routed through network interfaces that are not part of the configured network domain, the packets are still forwarded, but their QoS and queuing behavior is based on default settings. Also, the packet does not appear in SAP statistics.

There is always one network-domain with the reserved name default. The interfaces always belong to a default network-domain. It is possible to assign a specific interface to different user-defined network-domains. The loopback and system interfaces are also associated with the default network-domain at the creation. However, any attempt to associate those interfaces with any explicitly defined network-domain is blocked at the CLI level because there is no benefit for that association.

Any SDP can be assigned only to one network domain. If none is specified, the system assigns the default network-domain. This means that all SAPs in VPLS have queues reaching all fwd-complexes serving interfaces that belong to the same network-domains as the SDPs.

It is possible to assign or remove network-domain association of the interface/SDP without requiring deletion of the respective object.

#### 2.1.1.3 System interface

The system interface is associated with a network entity (such as a specific router or switch), not a specific interface. The system interface is also referred to as the loopback address. The system interface is associated during the configuration of the following entities:

- termination point of service tunnels
- hops when configuring MPLS paths and LSPs
- addresses on a target router for BGP and LDP peering

The system interface is used to preserve connectivity (when routing reconvergence is possible) when an interface fails or is removed. The system interface is used as the router identifier, and a system interface must have an IP address with a 32-bit subnet mask.

#### 2.1.1.4 Unicast reverse path forwarding check

Unicast reverse path forwarding check (uRPF) helps to mitigate problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of denial-of-service (DoS) attacks, including smurf and tribe flood network (TFN), can take advantage of forged or rapidly changing source addresses to allow attackers to thwart efforts to locate or filter the attacks. For Internet service providers (ISPs) that provide public access, uRPF deflects such attacks by forwarding only packets with source addresses that are valid and consistent with the IP routing table. This action protects the network of the ISP, its customer, and the rest of the Internet.

uRPF is supported for both IPv4 and IPv6 on network and access. It is supported on any IP interface, including base router, IES, VPRN, and subscriber group interfaces.

In strict mode, uRPF checks whether the incoming packet has a source address that matches a prefix in the routing table, and whether the interface expects to receive a packet with this source address prefix.

In loose mode, uRPF checks whether the incoming packet has a source address that matches a prefix in the routing table; loose mode does not check whether the interface expects to receive a packet with a specific source address prefix.

Loose mode uRPF check is supported for ECMP, IGP shortcuts, and VPRN MP-BGP routes. Packets coming from a source that matches any ECMP, IGP shortcut, or VPRN MP-BGP route passes the uRPF check even when uRPF is set to strict mode on the incoming interface.

In the case of ECMP, this allows a packet received on an IP interface configured in strict uRPF mode to be forwarded if the source address of the packet matches an ECMP route, even if the IP interface is not a next-hop of the ECMP route or not a member of any ECMP routes. The strict-no-ecmp uRPF mode may be configured on any interface that is known to not be a next-hop of any ECMP route. When a packet is received on this interface, and the source address matches an ECMP route, the packet is dropped by uRPF.

If there is a default route, the following is included in the uRPF check:

- A loose mode uRPF check always succeeds.
- A strict mode uRPF check only succeeds if the source address matches any route (including the default route) where the next-hop is on the incoming interface for the packet.

Otherwise, the uRPF check fails.

If the source IP address matches a discard/blackhole route, the packet is treated as if it failed the uRPF check.

### 2.1.1.5 QoS policy propagation using BGP

This section describes QoS policy propagation using BGP (QPPB) as it applies to VPRN, IES, and router interfaces. see the "Internet Enhanced Service" section in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Guide: IES and VPRN* and the "IP Router Configuration" section in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Router Configuration Guide*.

QoS policy propagation using BGP (QPPB) is a feature that allows a route to be installed in the routing table with a forwarding-class and priority so that packets matching the route can receive the associated QoS. The forwarding-class and priority associated with a BGP route are set using BGP import route policies. This feature is called QPPB, even though the feature name refers to BGP specifically. On SR OS, QPPB is supported for BGP (IPv4, IPv6, VPN-IPv4, VPN-IPv6), RIP, and static routes.

SAP ingress and network QoS policies can achieve the same result as QPPB (for example, by assigning a packet arriving on an IP interface to a specific forwarding-class and priority/profile, based on the source address or destination address of the packet). However, the effort involved in creating the QoS policies, keeping them up-to-date, and applying them across many nodes is much greater than with QPPB. In a typical application of QPPB, a BGP route is advertised with a BGP community attribute that conveys a specific QoS. Routers that receive the advertisement accept the route into their routing table and set the forwarding-class and priority of the route from the community attribute.

#### 2.1.1.5.1 QPPB applications

There are two typical applications of QPPB:

- coordination of QoS policies between different administrative domains
- traffic differentiation within a single domain, based on route characteristics

#### 2.1.1.5.2 Inter-AS coordination of QoS policies

The user of an administrative domain "A" can use QPPB to signal to a peer administrative domain "B" that traffic sent to specific prefixes advertised by domain A should receive a specific QoS treatment in domain B. For example, an ASBR of domain A can advertise a prefix to domain B and include a BGP community attribute with the route. The community value implies a specific QoS treatment, as agreed by the two domains (in their peering agreement or service level agreement, for example). When the ASBR and other routers in domain B accept and install the route for that prefix into their routing table, they apply a QoS policy on selected interfaces that classifies traffic toward that prefix into the QoS class implied by the BGP community value.

QPPB may also be used to request that traffic sourced from specific networks receive appropriate QoS handling in downstream nodes that may span different administrative domains. This can be achieved by advertising the source prefix with a BGP community, as described. However, in this case, other approaches are equally valid, such as marking the DSCP or other CoS fields based on the source IP address, so that downstream domains can act based on a common understanding of the QoS treatment implied by different DSCP values.

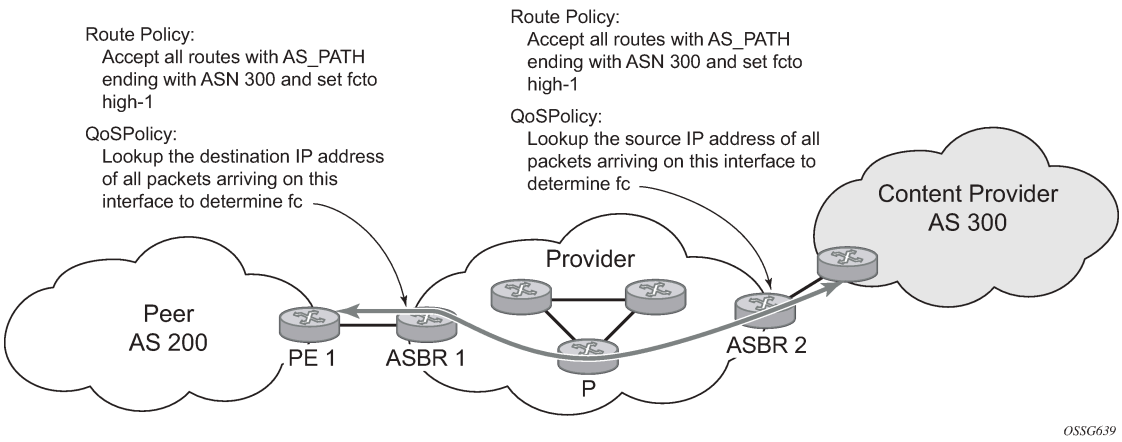
In the preceding examples, coordination of QoS policies using QPPB could be between a business customer and their IP VPN service provider, or between one service provider and another.

#### 2.1.1.5.3 Traffic differentiation based on route characteristics

A network user may need to provide differentiated service to specific traffic flows within its network, and these traffic flows can be identified with known routes. For example, the user of an ISP network may need to give priority to traffic originating in a specific ASN (the ASN of a content provider offering over-the-top services to the ISP's customers), following a specific AS_PATH, or destined for a specific next-hop (remaining on-net vs. off-net).

Figure 1: Use of QPPB to differentiate traffic in an ISP network shows an example of an ISP that has an agreement with the content provider managing AS300 to provide traffic sourced and terminating within AS300 with differentiated service appropriate to the content being transported. In this example, ASBR1 and ASBR2 mark the DSCP of packets terminating and sourced, respectively, in AS300 so that other nodes within the ISP's network do not need to rely on QPPB to determine the correct forwarding-class to use for the traffic. The DSCP or other CoS markings could be left unchanged in the ISP's network and QPPB used on every node.

*Figure 1: Use of QPPB to differentiate traffic in an ISP network*




### 2.1.1.6 QPPB

There are two main aspects of the QPPB feature:

- The ability to associate a forwarding-class and priority with specific routes in the routing table.
- The ability to classify an IP packet arriving on a specific IP interface to the forwarding-class and priority associated with the route that best matches the packet.

#### 2.1.1.6.1 Associating an FC and priority with a route

This feature uses the **fc** command in the route-policy hierarchy to set the forwarding class and, optionally, the priority associated with routes accepted by a route-policy entry.

The following example shows the **fc** command in the route policy configuration.

**Example: MD-CLI**

```
[ex:/configure policy-options]
A:admin@node-2# info
    community "gold" {
        member "300:100" { }
    }
    policy-statement "qppb_policy" {
        entry 10 {
            from {
                community {
                    name "gold"
                }
                protocol {
                    name [bgp]
                }
            }
            action {
                action-type accept
                forwarding-class {
                    fc h1
                    priority high
                }
```

```
            }
        }
    }
```

### Example: classic CLI

```
A:node-2>config>router>policy-options# info
----------------------------------------------
    community gold members 300:100
    policy-statement qppb_policy
        entry 10
            from
                protocol bgp
                community gold
            exit
            action accept
                fc h1 priority high
            exit
        exit
    exit
```

The **fc** command is supported with all existing from and to match conditions in a route policy entry, with any action other than reject, and with next-entry, next-policy, and accept actions. If a next-entry or next-policy action results in multiple matching entries, then the last entry with a QPPB action determines the forwarding class and priority.

A route policy that includes the **fc** command in one or more entries can be used in any import or export policy, but the **fc** command has no effect except in the following contexts:

- **MD-CLI**
  - VRF import policies

    ```
    configure service vprn bgp-evpn mpls vrf-import
    configure service vprn bgp-ipvpn mpls vrf-import
    configure service vprn bgp-ipvpn srv6 vrf-import
    configure service vprn mvpn vrf-import
    ```

  - BGP import policies:

    ```
    configure router bgp import
    configure router bgp group import
    configure router bgp neighbor import
    configure service vprn bgp import
    configure service vprn bgp group import
    configure service vprn bgp neighbor import
    ```

  - RIP import policies

    ```
    configure router rip import-policy
    configure router rip group import-policy
    configure router rip group neighbor import-policy
    configure service vprn rip import-policy
    configure service vprn rip group import-policy
    configure service vprn rip group neighbor import-policy
    ```

- **classic CLI**

– VRF import policies

```
configure service vprn bgp-evpn mpls vrf-import
configure service vprn bgp-ipvpn mpls vrf-import
configure service vprn bgp-ipvpn srv6 vrf-import
configure service vprn mvpn vrf-import
```

– BGP import policies

```
configure router bgp import
configure router bgp group import
configure router bgp group neighbor import
configure service vprn bgp import
configure service vprn bgp group import
configure service vprn bgp group neighbor import
```

– RIP import policies

```
configure router rip import
configure router rip group import
configure router rip group neighbor import
configure service vprn rip import
configure service vprn rip group import
configure service vprn rip group neighbor import
```

As shown, QPPB route policies support routes learned from RIP and BGP neighbors of a VPRN, as well as for routes learned from RIP and BGP neighbors of the base/global routing instance.

QPPB is supported for BGP routes belonging to any of the following address families:

- IPv4 (AFI=1, SAFI=1)
- IPv6 (AFI=2, SAFI=1)
- VPN-IPv4 (AFI=1, SAFI=128)
- VPN-IPv6 (AFI=2, SAFI=128)

A VPN-IP route may match both a VRF import policy entry and a BGP import policy entry (if **vpn-apply-import** is configured in the base router BGP instance). In this case, the VRF import policy is applied first, then the BGP import policy, so the QPPB QoS is based on the BGP import policy entry.

This feature also provides the ability to associate a forwarding-class and, optionally, priority with IPv4 and IPv6 static routes. This is achieved by specifying the forwarding class within the commands in the following contexts. Use the commands in the following contexts to configure the forwarding class.

- **MD-CLI**

```
configure router static-routes route next-hop qos
configure service vprn static-routes route next-hop qos
configure router static-routes route indirect qos
configure service vprn static-routes route indirect qos
```

- **classic CLI**

```
configure router static-route-entry next-hop
configure service vprn static-route-entry next-hop
configure router static-route-entry indirect
configure service vprn static-route-entry indirect
```

Priority is optional when specifying the forwarding class of a static route, but when configured it can only be deleted and returned to unspecified by deleting the entire static route.

### 2.1.1.6.2 Displaying QoS information associated with routes

Use the commands in the following contexts to show the forwarding class and priority associated with the displayed routes.

```
show router route-table
show router fib
show router bgp routes
show router rip database
show router static-route
```

Use the following command to show an additional line per route entry that displays the forwarding class and priority of the route. When the **qos** command option is specified, the output includes an additional line per route entry that displays the forwarding class and priority of the route. If a route has no forwarding class and priority information, the third line is blank.

```
show router route-table 10.1.5.0/24 qos
```

The following example shows the output of this command.

**Output example**

```
===============================================================================
Route Table (Router: Base)
===============================================================================
Dest Prefix                                 Type    Proto    Age         Pref
       Next Hop[Interface Name]                                  Metric
       QoS
-------------------------------------------------------------------------------
10.1.5.0/24                                 Remote  BGP      15h32m52s   0
       PE1_to_PE2                                             0
       h1, high
-------------------------------------------------------------------------------
No. of Routes: 1
===============================================================================
```

### 2.1.1.6.3 Enabling QPPB on an IP interface

To enable QoS classification of ingress IP packets on an interface based on the QoS information associated with the routes that best match the packets, configure the **qos-route-lookup** command in the IP interface. The **qos-route-lookup** command has command options to indicate whether the QoS result is based on lookup of the source or destination IP address in every packet. There are separate **qos-route-lookup** commands for the IPv4 and IPv6 packets on an interface, which allows QPPB to be enabled for IPv4 only, IPv6 only, or both IPv4 and IPv6. Currently, QPPB based on a source IP address is not supported for IPv6 packets or for ingress subscriber management traffic on a group interface.

The **qos-route-lookup** command is supported on the following types of IP interfaces:

- base router network interfaces

```
configure router interface
```

- VPRN SAP and spoke SDP interfaces

```
configure service vprn interface
```

- VPRN group-interfaces

```
configure service vprn subscriber-interface group-interface
```

- IES SAP and spoke SDP interfaces

```
configure service ies interface
```

- IES group interfaces

```
configure service ies subscriber-interface group-interface
```

When the **qos-route-lookup** command with the **destination** command option is applied to an IP interface and the destination address of an incoming IP packet matches a route with QoS information, the packet is classified to the FC and priority associated with that route. The command overrides the FC and priority/profile determined from the SAP ingress or network QoS policy associated with the IP interface. If the destination address of the incoming packet matches a route with no QoS information, the FC and priority of the packet remain as determined by the SAP ingress or network QoS policy.

When the **qos-route-lookup** command with the **source** command option is applied to an IP interface and the source address of an incoming IP packet matches a route with QoS information, the packet is classified to the FC and priority associated with that route. The command overrides the FC and priority/profile determined from the SAP ingress or network QoS policy associated with the IP interface. If the source address of the incoming packet matches a route with no QoS information, the FC and priority of the packet remain as determined by the SAP ingress or network QoS policy.

When the **qos-route-lookup** command is configured with the **source-and-dest** option (only available on systems with FP4 or a later generation FP datapath), the FC and priority of incoming packets is based on the lookup of the destination IP address, as long as the route matching the destination address has QPPB information. Otherwise, the FC and priority of incoming packets is based on the lookup of the source IP address, as long as the route matching the source address has QPPB information. If neither the destination nor the source lookups find a route with QPPB information, the FC and priority of the packet remain as determined by the SAP ingress or network QoS policy.

Currently, QPPB is not supported for ingress MPLS traffic on CsC PE'-CE' interfaces or on network interfaces, such as the following context.

```
configure service vprn network-interface
```

**Note:**
QPPB based on a source IP address is not supported for ingress subscriber management traffic on a group interface.

### 2.1.1.6.4 QPPB when next-hops are resolved by QPPB routes

In some cases (IP VPN inter-AS model C, Carrier Supporting Carrier, indirect static routes, and so on), an IPv4 or IPv6 packet may arrive on a QPPB-enabled interface and match a route A1 whose next-hop N1 is resolved by a route A2 with next-hop N2. Similarly, N2 is resolved by a route A3 with next-hop N3, and so on. The QPPB result is based only on the forwarding-class and priority of route A1. If A1 does not have a

forwarding-class and priority association, the QoS classification is not based on QPPB, even if routes A2, A3, and so on, have forwarding-class and priority associations.

#### 2.1.1.6.5 QPPB and multiple paths to a destination

When ECMP is enabled, some routes may have multiple equal-cost next-hops in the forwarding table. When an IP packet matches such a route, the next-hop selection is typically based on a hash algorithm that tries to load balance traffic across all the next-hops while keeping all packets of a flow on the same path. The QPPB configuration model described in Associating an FC and priority with a route allows different QoS information to be associated with the different ECMP next-hops of a route. The forwarding-class and priority of a packet matching an ECMP route is based on the next-hop used to forward the packet.

When Edge PIC [1] is enabled, some BGP routes may have a backup next-hop in the forwarding table, as well as the one or more primary next-hops representing the equal-cost best paths allowed by the ECMP/multipath configuration. When an IP packet matches such a route, a reachable primary next-hop is selected (based on the hash result) but if all the primary next-hops are unreachable, the backup next-hop is used. The QPPB configuration model described in Associating an FC and priority with a route allows the forwarding-class and priority associated with the backup path to be different from the QoS characteristics of the equal-cost best paths. The forwarding class and priority of a packet forwarded on the backup path is based on the FC and priority of the backup route.

#### 2.1.1.6.6 QPPB and policy-based routing

When an IPv4 or IPv6 packet with destination address arrives on an interface with both QPPB and policy-based-routing enabled:

- There is no QPPB classification if the IP filter action redirects the packet to a directly connected interface, even if the destination address is matched by a route with a forwarding-class and priority.
- QPPB classification is based on the forwarding-class and priority of the route matching IP address Y if the IP filter action redirects the packet to the indirect next-hop IP address Y, even if the destination address is matched by a route with a forwarding-class and priority.

### 2.1.1.7 QPPB and GRT lookup

Source-address based QPPB is not supported on any SAP or spoke SDP interface of a VPRN configured with the **grt-lookup** command.

#### 2.1.1.7.1 QPPB interaction with SAP ingress QoS policy

When QPPB is enabled on a SAP IP interface, the forwarding class of a packet may change from **fc1** (the original FC determined by the SAP ingress QoS policy) to fc2, the new FC determined by QPPB. In the ingress datapath, SAP ingress QoS policies are applied in the first P chip and route lookup/QPPB occurs in the second P chip. This has the following implications:

- Ingress remarking (based on profile state) is always based on the original FC (fc1) and sub-class (if defined).
- The profile state of a SAP ingress packet that matches a QPPB route depends on the configuration of **fc2** only. If the de-1-out-profile flag is enabled in **fc2**, and **fc2** is not mapped to a priority mode queue,

the packet is marked out of profile if its DE bit = 1. If the profile state of **fc2** is explicitly configured (in or out) and **fc2** is not mapped to a priority mode queue, the packet is assigned this profile state. In both cases, there is no consideration of whether **fc1** was mapped to a priority mode queue.

- The priority of a SAP ingress packet that matches a QPPB route depends on several factors. If the de-1-out-profile flag is enabled in **fc2** and the DE bit is set in the packet, priority is low regardless of the QPPB priority or **fc2** mapping to profile mode queue, priority mode queue, or policer. If **fc2** is associated with a profile mode queue, the packet priority is based on the explicitly configured profile state of **fc2** (in profile = high, out profile = low, undefined = high), regardless of the QPPB priority or **fc1** configuration. If **fc2** is associated with a priority mode queue or policer, the packet priority is based on QPPB (unless DE=1). If no priority information is associated with the route, the packet priority is based on the configuration of **fc1**. If **fc1** mapped to a priority mode queue, the priority is based on DSCP/IP prec/802.1p. If **fc1** mapped to a profile mode queue, the priority is based on the profile state of **fc1**.

Table 1: QPPB interactions with SAP ingress QoS summarizes these interactions.

*Table 1: QPPB interactions with SAP ingress QoS*

| Original FC object mapping | New FC object mapping | Profile | Priority (drop preference) | DE=1 override | In/out of profile marking |
|---|---|---|---|---|---|
| Profile mode queue | Profile mode queue | From new base FC unless overridden by DE=1 | From QPPB, unless packet is marked in or out of profile in which case follows profile Default: high priority | From new base FC | From original FC and sub-class |
| Priority mode queue | Priority mode queue | Ignored | If DE=1 override then low otherwise from QPPB, if no DEI or QPPB overrides then from original dot1p/ exp/DSCP mapping or policy default | From new base FC | From original FC and sub-class |
| Policer | Policer | From new base FC unless overridden by DE=1 | If DE=1 override then low otherwise from QPPB, if no DEI or QPPB overrides then from original dot1p/ exp/DSCP mapping or policy default | From new base FC | From original FC and sub-class |
| Priority mode queue | Policer | From new base FC unless overridden by DE=1 | If DE=1 override then low otherwise from QPPB, if no DEI or QPPB overrides then from original dot1p/ exp/DSCP mapping or policy default | From new base FC | From original FC and sub-class |

| Original FC object mapping | New FC object mapping | Profile | Priority (drop preference) | DE=1 override | In/out of profile marking |
|---|---|---|---|---|---|
| Policer | Priority mode queue | Ignored | If DE=1 override then low otherwise from QPPB, if no DEI or QPPB overrides then from original dot1p/ exp/DSCP mapping or policy default | From new base FC | From original FC and sub-class |
| Profile mode queue | Priority mode queue | Ignored | If DE=1 override then low otherwise from QPPB, if no DEI or QPPB overrides then follows original FC's profile mode rules | From new base FC | From original FC and sub-class |
| Priority mode queue | Profile mode queue | From new base FC unless overridden by DE=1 | From QPPB, unless packet is marked in or out of profile in which case follows profile Default: high priority | From new base FC | From original FC and sub-class |
| Profile mode queue | Policer | From new base FC unless overridden by DE=1 | If DE=1 override then low otherwise from QPPB, if no DEI or QPPB overrides then follows original FC's profile mode rules | From new base FC | From original FC and sub-class |
| Policer | Profile mode queue | From new base FC unless overridden by DE=1 | From QPPB, unless packet is marked in or out of profile in which case follows profile Default: high priority | From new base FC | From original FC and sub-class |

### 2.1.1.8 Configuring link delay

The delay represents the unidirectional link delay from the local router to the remote router (that is, the forward-path latency). The interface delay is a link property and is typically calculated as the combination of speed of light versus fiber length versus fiber composition. Typically, these delay components are not subject to sudden change in a network. If a change occurs, it may be caused by fiber cuts (such as light out), or Layer 1 reroute events.

If delay is configured for all links in the network, the attribute can be used as a feasible metric for SR flex-algo applications.

The static delay represents a forward-path metric, in microseconds, between two routers. It is not possible to configure a delay on a loopback or system interface; the delay IGP extension TLVs (specified in RFC

8570) are not defined for stub links. The delay is encoded in IGP application-specific attributes (for example, for IS-IS, see *draft-ietf-isis-te-app-14.txt*). The delay can be configured upon other interface links.

The default setting is no delay, which means that IGP (for example, IS-IS) does not add a link delay metric TLV. The lack of this TLV in flex-algo causes the link with the no delay TLV setting to be pruned from the topology.

The following example shows the configuration of link delay.

**Output example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
    interface interface-name {
        if-attribute {
            delay {
                static microseconds
            }
        }
    }
```

**Output example: classic CLI**

```
A:node-2>config>router# info
#------------------------------------------------
echo "IP Configuration"
#------------------------------------------------
        interface "interface-name"
            if-attribute
                delay
                    static microseconds
                exit
            exit
            no shutdown
        exit
```

The static delay can be configured within the range 1 to 16777214 microseconds.

## 2.1.2 Router ID

The router ID, a 32-bit number, uniquely identifies the router within an autonomous system (AS) (see Autonomous systems). In protocols such as OSPF, routing information is exchanged between areas—groups of networks that share routing information. It can be set to be the same as the loopback address. The router ID is used by both OSPF and BGP routing protocols in the routing table manager instance.

There are several ways to obtain the router ID. On each router, the router ID can be obtained in the following ways.

- Define the router ID. Use the following command to define the value that becomes the router ID.

  ```
  configure router
  ```

- Configure the system interface with an IP address. If the router ID is not manually configured in the **configure router** context, the system interface acts as the router ID. Use the following command to configure the system interface with an IP address.

  ```
  configure router interface
  ```

- If neither the system interface nor router ID are implicitly specified, the router ID is inherited from the last four bytes of the MAC address.
- The router can be obtained from the protocol level; for example, BGP.

### 2.1.3 Autonomous systems

Networks can be grouped into areas. An area is a collection of network segments within an autonomous system (AS) that have been administratively assigned to the same group. An area's topology is concealed from the rest of the AS, which results in a significant reduction in routing traffic.

Routing in the AS takes place on two levels, depending on whether the source and destination of a packet reside in the same area (intra-area routing) or different areas (inter-area routing). In intra-area routing, the packet is routed solely on information obtained within the area; no routing information obtained from outside the area can be used. This protects intra-area routing from the injection of bad routing information.

Routers that belong to more than one area are called area border routers. All routers in an AS do not have an identical topological database. An area border router has a separate topological database for each area it is connected to. Two routers, which are not area border routers, belonging to the same area, have identical area topological databases.

Autonomous systems share routing information, such as routes to each destination and information about the route or AS path, with other ASes using BGP. Routing tables contain lists of next hops, reachable addresses, and associated path cost metrics to each router. BGP uses the information and path attributes to compile a network topology.

### 2.1.4 Confederations

Configuring confederations is optional and should only be implemented to reduce the interior border gateway protocol (IBGP) mesh inside an AS. An AS can be logically divided into smaller groupings called sub-confederations and then assigned a confederation ID (similar to an autonomous system number). Each sub-confederation has fully meshed IBGP and connections to other ASs outside of the confederation.

The sub-confederations have EBGP-type peers to other sub-confederations within the confederation. They exchange routing information as if they were using IBGP. Command options such as next hop, metric, and local preference are preserved. The confederation appears and behaves like a single AS.

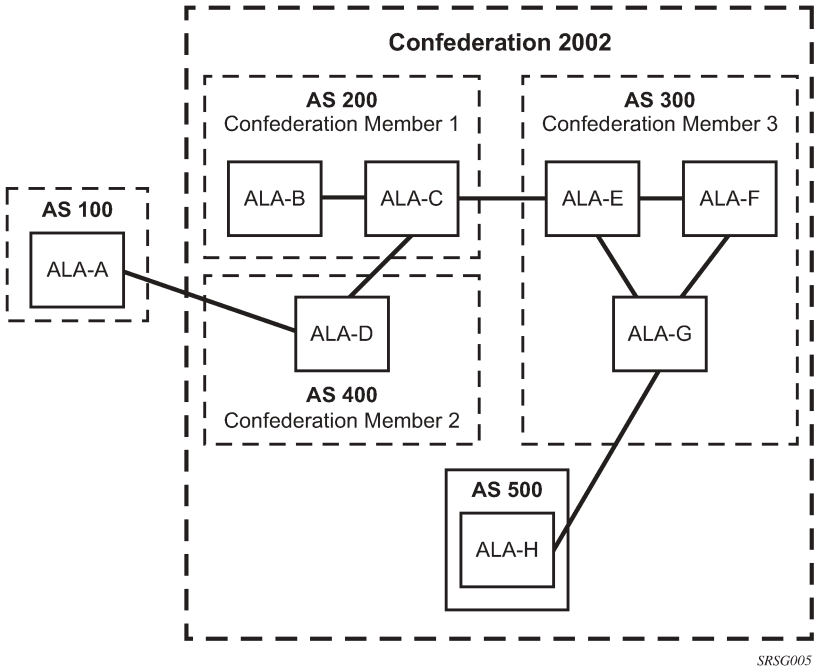
Confederations have the following characteristics:

- A large AS can be sub-divided into sub-confederations.
- Routing within each sub-confederation is accomplished via IBGP.
- EBGP is used to communicate between sub-confederations.
- BGP speakers within a sub-confederation must be fully meshed.
- Each sub-confederation (member) of the confederation has a different AS number. The AS numbers used are typically in the private AS range of 64512 to 65535.

To migrate from a non-confederation configuration to a confederation configuration requires a major topology change and configuration modifications on each participating router. Setting BGP policies to select an optimal path through a confederation requires other BGP modifications.

There are no default confederations. Router confederations must be explicitly created. Figure 2: Confederation configuration shows an example of a confederation configuration.

*Figure 2: Confederation configuration*




## 2.1.5 Proxy ARP

Proxy ARP is the technique in which a router answers ARP requests intended for another node. The router appears to be present on the same network as the "real" node that is the target of the ARP and takes responsibility for routing packets to the "real" destination. Proxy ARP can help nodes on a subnet reach remote subnets without configuring routing or a default gateway.

Typical routers only support proxy ARP for directly attached networks; the router is targeted to support proxy ARP for all known networks in the routing instance where the virtual interface proxy ARP is configured.

To support DSLAM and other edge-like environments, proxy ARP supports policies that allow the provider to configure prefix lists that determine for which target networks proxy ARP is attempted and prefix lists that determine for which source hosts proxy ARP is attempted.

Also, the proxy ARP implementation supports the ability to respond for other hosts within the local subnet domain. This is needed in environments such as DSL where multiple hosts are in the same subnet but cannot reach each other directly.

Static ARP is used when a Nokia router needs to know about a device on an interface that cannot or does not respond to ARP requests. The configuration can state that, if it has a packet with a specific IP address, to send it to the corresponding ARP address. Use proxy ARP so the router responds to ARP requests on behalf of another device.

### 2.1.6 Exporting an inactive BGP route from a VPRN

Use the following command to provide an IP VPN command option that allows the best BGP route learned by a VPRN to be exported as a VPN-IP route even when that BGP route is inactive because of the presence of a more preferred BGP-VPN route from another PE.

```
configure service vprn export-inactive-bgp
```

This "best-external" type of route advertisement is useful in active or standby multihoming scenarios because it can ensure that all PEs have knowledge of the backup path provided by the standby PE.

### 2.1.7 DHCP relay

See the *7450 ESS, 7750 SR, and VSR Triple Play Service Delivery Architecture Guide* for information about DHCP relay and support, as well as configuration examples.

### 2.1.8 Internet protocol versions

The -SR OS implements IP routing functionality, providing support for IP version 4 (IPv4) and IP version 6 (IPv6). IP version 6 (RFC 1883, *Internet Protocol, Version 6 (IPv6)*) is a version of the Internet Protocol designed as a successor to IP version 4 (IPv4) (RFC-791, *Internet Protocol*). The changes from IPv4 to IPv6 affect the following categories:

- **expanded addressing capabilities**

  IPv6 increases the IP address size from 32 bits (IPv4) to 128 bits, to support more levels of addressing hierarchy, a much greater number of addressable nodes, and simpler auto-configuration of addresses. The scalability of multicast routing is improved by adding a scope field to multicast addresses. Also, a type of address called an anycast address is defined that is used to send a packet to any one of a group of nodes.

- **header format simplification**

  Some IPv4 header fields have been dropped or made optional to reduce the common-case processing cost of packet handling and to limit the bandwidth cost of the IPv6 header.

- **improved support for extensions and options**

  Changes in the way IP header options are encoded allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing options in the future.

- **flow labeling capability**

  The capability to enable the labeling of packets belonging to traffic flows for which the sender requests special handling, such as non-default quality of service or "real-time" service was added in IPv6.

- **authentication and privacy capabilities**

  Extensions to support authentication, data integrity, and (optional) data confidentiality are specified for IPv6.

*Figure 3: IPv6 header format*

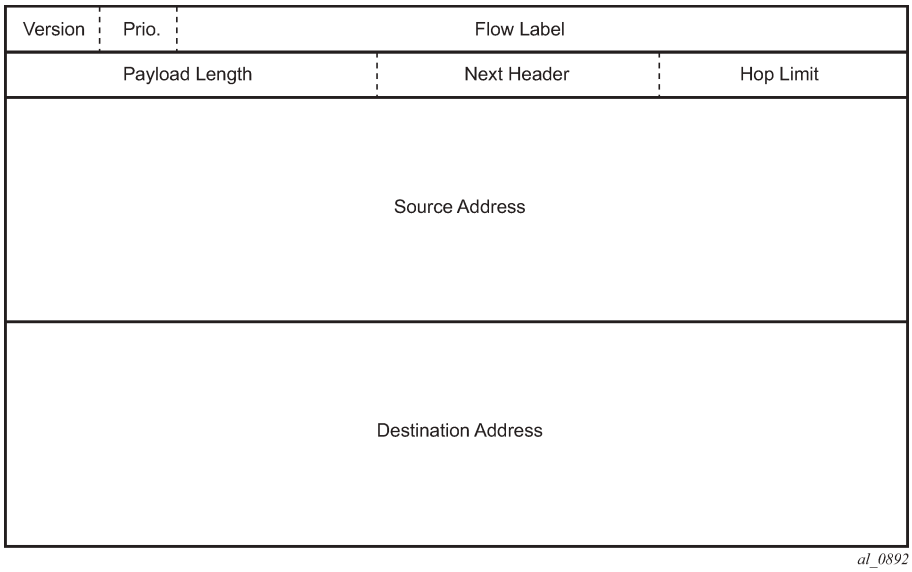



*Table 2: IPv6 header field descriptions*

| Field | Description |
|---|---|
| Version | 4-bit Internet Protocol version number = 6 |
| Prio. | 4-bit priority value |
| Flow Label | 24-bit flow label |
| Payload Length | 16-bit unsigned integer; the length of payload, for example, the rest of the packet following the IPv6 header, in octets; if the value is zero, the payload length is carried in a jumbo payload hop-by-hop option |
| Next Header | 8-bit selector; identifies the type of header immediately following the IPv6 header; this field uses the same values as the IPv4 protocol field |
| Hop Limit | 8-bit unsigned integer; decremented by 1 by each node that forwards the packet; the packet is discarded if the hop limit is decremented to zero |
| Source Address | 128-bit address of the originator of the packet. |
| Destination Address | 128-bit address of the intended recipient of the packet (possibly not the ultimate recipient if a routing header is present) |

### 2.1.8.1 IPv6 address format

IPv6 uses a 128-bit address, as opposed to the IPv4 32-bit address. Unlike IPv4 addresses, which use the dotted-decimal format, with each octet assigned a decimal value from 0 to 255, IPv6 addresses use the colon-hexadecimal format X:X:X:X:X:X:X:X, where each X is a 16-bit section of the 128-bit address. For example:

2001:0db8:0000:0000:0000:0000:0000:0000

Leading zeros must be omitted from each block in the address. A series of zeros can be replaced with a double colon. For example:

2001:db8::

The double colon can only be used one time in an address.

The IPv6 prefix is the part of the IPv6 address that represents the network identifier, which appears at the beginning of the address. The IPv6 prefix length, which begins with a forward slash (/), shows how many bits of the address make up the network identifier. For example, the address 2001:db8:8086:6502::1/64 means that the first 64 bits of the address represent the network identifier; the remaining 64 bits represent the node identifier.

**Note:**
IPv6 addresses and prefixes are displayed according to RFC 5952, *A Recommendation for IPv6 Address Text Representation*.

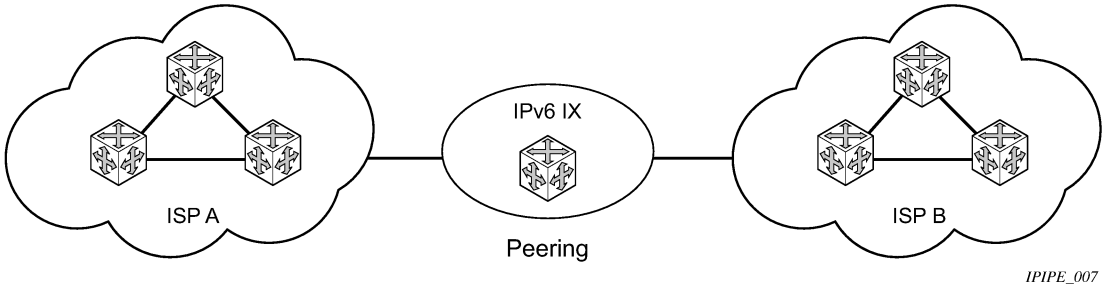
### 2.1.8.2 IPv6 applications

Examples of the IPv6 applications supported by the SR OS include:

- **IPv6 Internet exchange peering**

  Figure 4: IPv6 Internet exchange shows an IPv6 Internet exchange where multiple ISPs peer over native IPv6

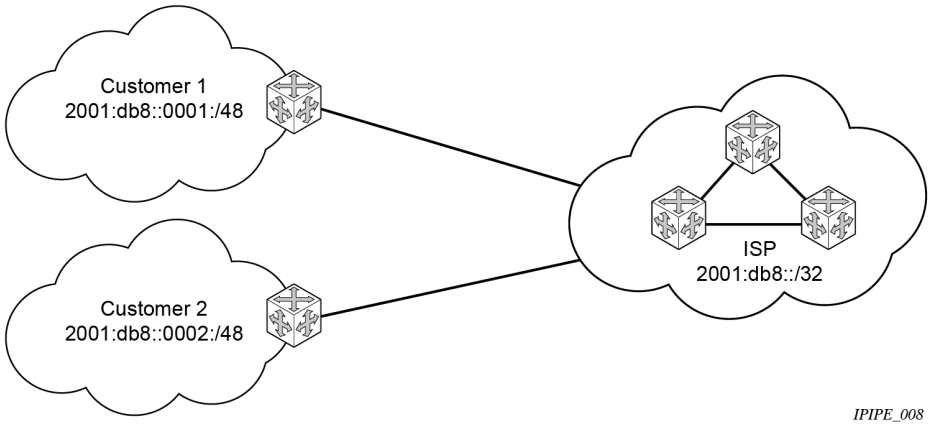
  *Figure 4: IPv6 Internet exchange*




- **IPv6 transit services**

  Figure 5: IPv6 transit services shows IPv6 transit services provided by an ISP.

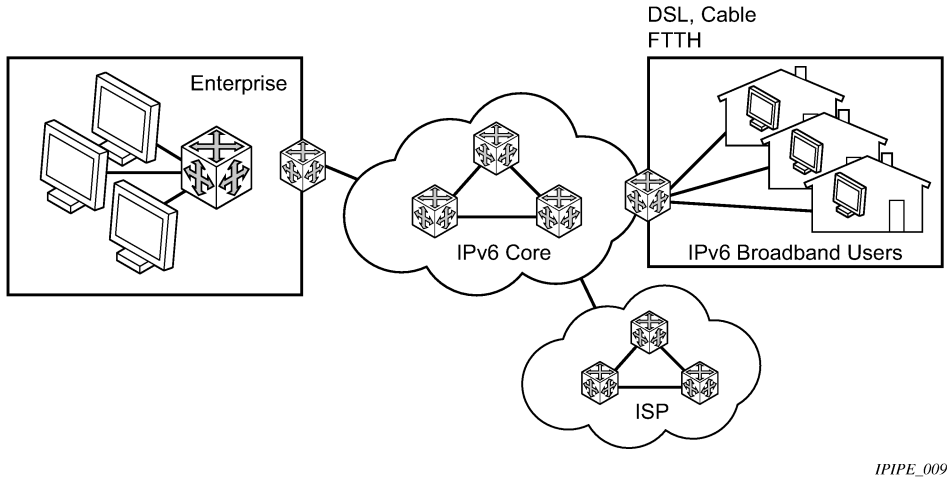
*Figure 5: IPv6 transit services*




- IPv6 services to enterprise customers and home users

  Figure 6: IPv6 services to enterprise customers and home users shows IPv6 services to enterprise and home broadband users.

  *Figure 6: IPv6 services to enterprise customers and home users*




- **IPv6 over IPv4 relay services**

  IPv6 over IPv4 tunnels are one of many IPv6 transition methods to support IPv6 in an environment where not only IPv4 exists but native IPv6 networks depend on IPv4 for greater IPv6 connectivity. Nokia routers support dynamic IPv6 over IPv4 tunneling. The IPv4 source and destination address are taken from configuration, the source address is the IPv4 system address and the IPv4 destination is the next hop from the configured IPv6 over IPv4 tunnel.

  IPv6 over IPv4 is an automatic tunnel method that gives a prefix to the attached IPv6 network. Figure 7: IPv6 over IPv4 tunnels shows IPv6 over IPv4 tunneling to transition from IPv4 to IPv6.

*Figure 7: IPv6 over IPv4 tunnels*

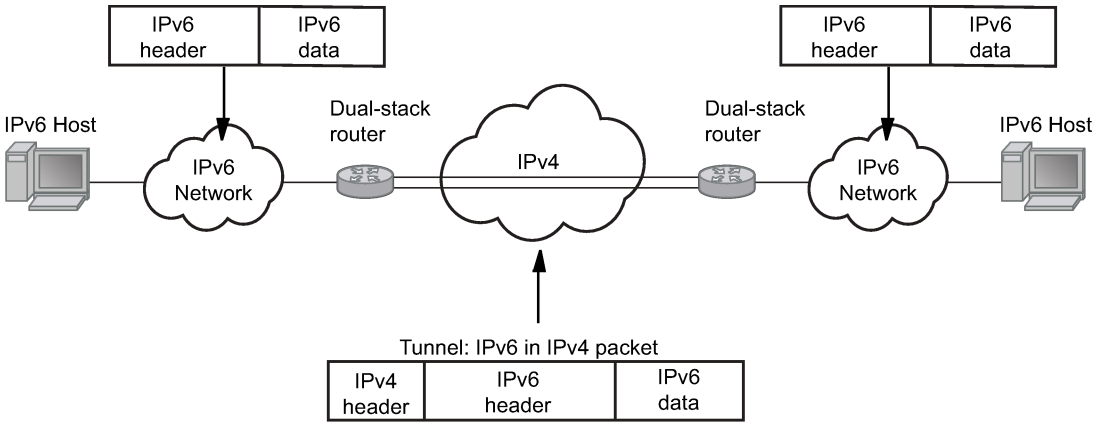



*Fig_29a*

### 2.1.8.3 DNS

The DNS client is extended to use IPv6 as transport and to handle the IPv6 address in the DNS AAA resource record from an IPv4 or IPv6 DNS server. An assigned name can be used instead of an IPv6 address because IPv6 addresses are more difficult to remember than IPv4 addresses.

#### 2.1.8.3.1 DNS resolution using a VPRN

When using a management VPRN, to allow DNS resolution via VPRN, as an example, DNS for all packets—routed through the Global Routing Table or the VPRN—the user must enable a redirect VPRN configuration under the base DNS server.

Use the following command to enable the redirect VPRN configuration.

```
configure router dns redirect-vprn service
```

When the redirect-vprn configuration is enabled, all packets have their URLs resolved through the configured redirect-vprn service. Only a single redirect-vprn configuration is supported.

As a prerequisite for the DNS resolution through the VPRN, the VPRN DNS server must be configured with at least a primary-dns IP address (IPv4 or IPv6). If the VPRN DNS server is not configured, all packet resolution fails, even if the BOF DNS server is configured, because the redirect-vprn configuration forces all packets through the redirect-vprn service for resolution.

The **redirect-vprn** command is not available at bootup, because the configuration is not loaded yet. Until the **redirect-vprn** command is executed, all DNS resolution is possible only through the BOF DNS configuration. The redirect-vprn configuration becomes active at runtime, after the configuration file is loaded and the **redirect-vprn** command is executed.

If the **redirect-vprn** command is not configured, DNS resolution occurs as follows:

- The global routing packets use the BOF DNS server.

- The VPRN packets use the configured VPRN DNS server. If the VPRN DNS server is not configured, the resolution occurs through the BOF DNS server.

For information about management VPRNs, see *Node Management Using VPRN* in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Guide: IES and VPRN*.

### 2.1.8.4 Secure Neighbor Discovery

Secure Neighbor Discovery (SeND) in conjunction with Cryptographically Generated Addresses (CGAs) allows users to secure IPv6 neighbor discovery between nodes on a common Layer 2 network segment.

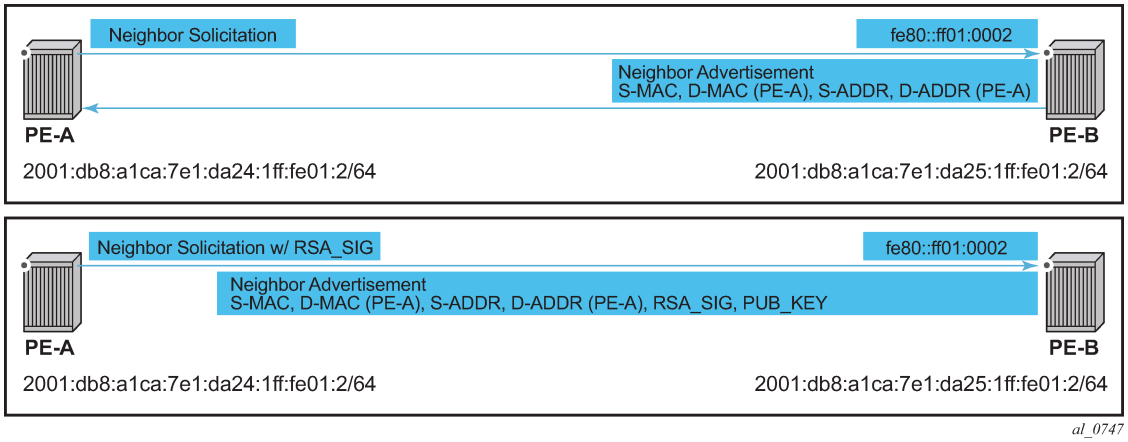
When SeND is enabled on an interface, CGAs must be enabled and static GUA/LLA IPv6 addressing is not supported. In this case, the router generates a CGA from the configured prefix (GUA, LLA) and use that address for all communication. The router validates NS/ND messages from other nodes on the network segment, and only install them in the neighbor cache if they pass validation.

Several potential use-cases for SeND exist to secure the network from deliberate or accidental tampering during neighbor discovery, SeND can prevent hijacking of in-use IPv6 addressing or man-in-the-middle attacks, but also to validate whether a node is permitted to participate in neighbor discovery, or validate which routers are permitted to act as default gateways.

SeND affects the following areas of neighbor discovery:

- Neighbor solicitation (solicited-node multicast address; target address)
- Neighbor advertisement (solicited; unsolicited)
- Router solicitation
- Router advertisement
- Redirect messages

*Figure 8: Neighbor discovery with and without SeND*




When SeND is enabled on a node, basic neighbor discovery messaging is changed as shown in Figure 8: Neighbor discovery with and without SeND. In the example, PE-A needs to find the MAC address of PE-B.

1. PE-A sends an NS message to the solicited node multicast address for PE-B's address with the CGA option, RSA signature option, timestamp option, and nonce option.

2. PE-B processes the NS message and, because it is configured for SeND operation, processes the NS. PE-B validates the source address of the packet to ensure it is a valid CGA, then validate the cryptographic signature embedded in the NS message.
3. PE-B generates an NA message, which is sent back to PE-A with the solicited bit, router bit set. The source address is that of PE-B, while the destination address is that of PE-A from the NS message. The timestamp is generated from PE-B, while the nonce is copied from PE-A's NS message.
4. PE-A receives the NA and completes similar checks as PE-B did.

If all steps process correctly, both nodes install each other’s addresses into their neighbor cache database.

### 2.1.8.5 SeND persistent CGAs

Persistent CGAs is a feature of SeND.

Previously, all generated CGAs on SeND-enabled interfaces remained unchanged after a CPM switchover, but after a reboot from a saved configuration file, all CGAs were regenerated.

To keep the same CGAs after a reboot from a saved configuration file:

1. Save the RSA key pair used for SeND.
2. Save the modifiers used during the CGA generation.

To make the CGAs persistent:

1. Import an online or offline generated RSA key pair for SeND.
2. Ensure that the CompactFlash (CF) files containing an RSA key pair that is used for SeND, are synchronized to the standby CPM by making use of the HA infrastructure used for certificates.
3. Ensure that the configuration file is saved when one or more CGAs are generated.

#### 2.1.8.5.1 Persistent RSA key pair

The RSA key pair is stored in a file on the CF.

##### Generate an RSA key pair

Use the following command to generate an RSA key pair:

- **MD-CLI**

```
admin system security pki generate-key-pair rsa-key-size
```

- **classic CLI**

```
admin certificate gen-keypair type rsa size
```

The following example shows the generation of an RSA key pair. This generates a Distinguished Encoding Rules (DER) formatted file.

##### Example: MD-CLI

```
[/admin system security pki]
A:admin@node-2# generate-keypair cf1:\myDir\myRsaKeyPair rsa-key-size 1024
```

### Example: classic CLI

```
*A:node-2# admin certificate gen-keypair cf1:\myDir\myRsaKeyPair type rsa size 1024
```

## Import an online or offline generated RSA key pair

In the classic CLI, use the following command to import a generated RSA key pair.

**Note:** The **secure-nd-import** command is only supported for the classic CLI.

```
admin certificate secure-nd-import
```

The following example shows the import of a generated RSA key pair.

### Example: classic CLI

```
*A:node-2# admin certificate secure-nd-import input cf1:\myDir\myRsaKeyPair output format
 der
```

Take the following into consideration when configuring the RSA key pairs:

- Because SeND only uses RSA key pairs, the command is refused if the imported key type is not RSA.
- Because SeND only supports key size 1024, the command is refused if the imported key size is not 1024.
- The password has to be specified when an offline generated file in pkcs12 format has to be imported.
- See the RSA key pair rollover mechanism section that follows for more information about key-rollover.
- In the classic CLI, use the following command to create the file cfx:\system-pki\secureNdKey (fixed directory and filename) and save the imported key in that file in encrypted per format same.

  ```
  admin certificate import
  ```

- The RSA key pair is uploaded in the memory of SeND.

## RSA key pair rollover mechanism

In the classic CLI, use the following command (described in the Import an online or offline generated RSA key pair section) to trigger a key rollover.

```
admin certificate secure-nd-import
```

The following example shows the triggering of a key rollover.

### Example: classic CLI

```
*A:node-2# admin certificate secure-nd-import cf1:\myDir\myOtherRsaKeyPair format der key-
rollover
```

Take the following into consideration when using the rollover mechanism:

- If CGAs exist that are generated based on an autogenerated or previously imported RSA key pair and the **key-rollover** keyword is not specified, the **secure-nd-import** command is refused.

- If a **secure-nd-import** with **key-rollover** is requested while a previous key rollover is still being handled, the new command is refused.
- If the **secure-nd-import** command is accepted, the imported RSA key pair is written to the file cfx:\system-pki\secureNdKey and loaded to SeND. Existing CGAs, if any, are regenerated.
- While handling a key rollover, SeND keeps track of which interface uses which RSA key pair. Temporarily, SeND can have two RSA key pairs in use. At all times, only the latest RSA key pair is stored in the file cfx:\system-pki\secureNdKey. When the rollover is finished, the RSA key pair that is no longer referred to, is deleted from SeND's memory.

### Autogeneration of an RSA key pair

The first time an interface becomes SeND enabled, SeND needs an RSA key pair to generate or check a modifier and to generate a CGA.

If the user did not import an RSA key pair for SeND, an autogenerated RSA key pair are used as a fallback.

The autogenerated RSA key pair is synchronized to the standby CPM, but is not written to the CF. Therefore, all CGAs generated via an autogenerated RSA key pair are not persistent. A warning is raised whenever a non-persistent CGA is generated.

In the classic CLI, the **admin certificate secure-nd-import** command without the key-rollover keyword is refused if CGAs exist that made use of the autogenerated RSA key pair. Specifying the **key-rollover** keyword results in regeneration of the CGAs.

See the section Making non-persistent CGAs persistent for more information about the procedure to make non-persistent CGAs persistent.

### HA

For the synchronization of the RSA key pair file in cfx:\system-pki\ used by SeND, use the following commands for manual and automatic certificate synchronization:

- **MD-CLI**
  Use the following command to manually synchronize:

  ```
  admin redundancy synchronize certificate
  ```

  Use the following command to automatically synchronize:

  ```
  configure redundancy cert-sync
  ```

- **classic CLI**
  Use the following command to manually synchronize:

  ```
  admin redundancy synchronize cert
  ```

  Use the following command to automatically synchronize:

  ```
  configure redundancy cert-sync
  ```

SeND also synchronizes the RSA key pair to the standby CPM.

### 2.1.8.5.2 Persistent CGA modifier

**Note:** This information applies to the classic CLI.

The modifier used during the CGA generation is saved in the configuration file. The CGA itself is not stored.

Based on the stored modifier and RSA key pair, the same CGA can be regenerated.

The modifier is needed to be sent out in ND messages.

By storing the modifier in the configuration file, the user can also configure an offline generated modifier (possibly with a security parameter > 1).

#### Example: Configure a SeND interface without modifiers (classic CLI)

```
A:node-2>config>router# info
#------------------------------------------------
  "IP Configuration"
#------------------------------------------------
        interface "itf1"
            address 10.10.10.1/24
            port 1/1/1
            ipv6
                secure-nd
                    no shutdown
```

A modifier is generated based on the actual RSA key pair (that is, imported or autogenerated). The offline modifier is used to generate a link-local CGA. The modifier is used to generate the global CGA.

```
exit
address 2000:1::/64
```

The modifier is saved in the interface configuration file.

#### Example: Configure a SeND interface with modifiers (classic CLI)

```
*A:node-2>config>router# info
...
#------------------------------------------------
echo "IP Configuration"
#------------------------------------------------
        interface "itf2"
            address 10.10.10.2/24
            port 1/1/2
            ipv6
                secure-nd
                    link-local-modifier 0xABCD
...
```

The offline modifier is used to generate the link-local CGA.

```
no shutdown
exit
address 3000:1::/64 modifier
```

A modifier is generated based on the actual RSA key pair. The modifier is used to generate the global CGA.

The modifier is stored in the interface configuration file.

**Example: Stored modifier (classic CLI)**

```
        address 3000:2::/64 modifier 0xABCD
```

The same offline generated modifier as the preceding link-local address is used for the generation of a global address.

**Example: Modifier for generation of a global address (classic CLI)**

```
address 3000:3::/64 modifier 0xABCD
```

Another offline generated modifier (*) is used for the generation of a global address.

For an offline generated modifier, a check is performed to see if it is generated with the actual RSA key pair and the security parameter applicable for the interface. If this check fails, the command is refused, unless the command is triggered in the context of an exec of a config file. In that case, the modifier is replaced by a new one that is generated based on the actual RSA key pair.

### 2.1.8.5.3 Making non-persistent CGAs persistent

**Note:** This information applies to the classic CLI.

CGAs can be non-persistent because:

- The user forgot to configure an RSA key pair for SeND, therefore, the CGAs were generated based on an auto-generated RSA key pair.
- The user forgot to synchronize an RSA key pair file to the stand-by CPM and a switchover happens.
- The CGAs were generated by a software version not having persistent CGAs (such as, ISSU).
- The system was booted from a configuration file generated by a software version not having persistent CGAs.

**Key rollover**

You can import a new RSA key pair for SeND with the **key-rollover** keyword. This results in the regeneration of all CGAs on all interfaces.

**Exporting the SeND RSA key pair**

Another method that does not result in the regeneration of the CGAs is to export the RSA key pair that is currently in use by SeND to the system-pki directory via an admin command.

In the classic CLI, use the following command to export the RSA key pair in use by SeND to the system-pki directory.

```
admin certificate secure-nd-import
```

This command writes the RSA key pair to the file cfx:\system-pki\secureNdKey in encrypted der format.

### 2.1.8.5.4 Booting from a saved configuration file

**Configuration saved by a software version with persistent CGAs**

The file cfx:\system-pki\secureNdKey should exist. This file is automatically uploaded by SeND during initialization.

The configuration file should contain a modifier for each address on a SeND enabled interface.

Modifiers in the configuration file are checked against the current RSA key pair. If the check fails, a new modifier and CGA is generated, and a warning is raised that a new CGA is generated.

If a modifier is missing from the configuration file for an IPv6 /64 prefix on a SeND enabled interface, a new modifier and CGA is generated based on the active RSA key pair.

**Configuration saved by a software version having non-persistent CGAs**

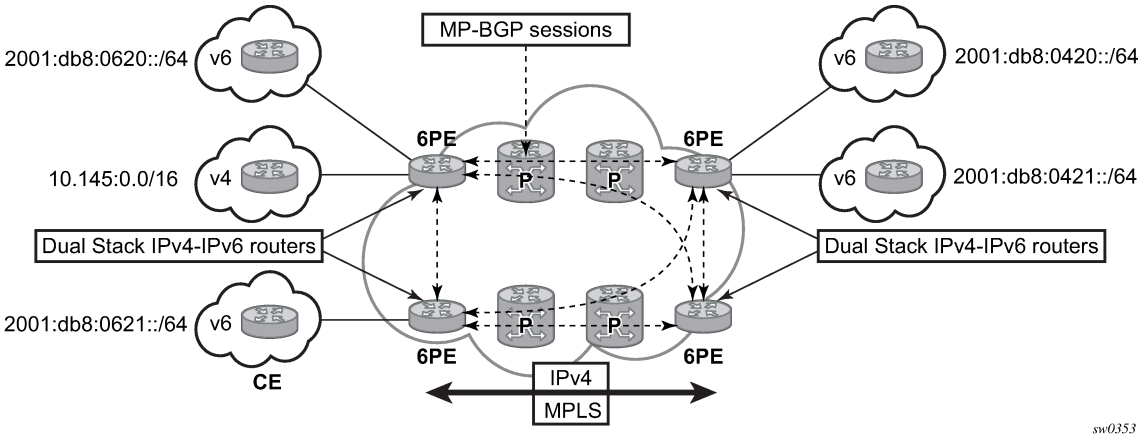
The file cfx:\system-pki\secureNdKey does not exist nor does the configuration file contain a modifier for any of the IPv6 /64 prefixes on secure-nd enabled interfaces.

New CGAs have to be generated (from the CLI context). Follow one of the procedures described in section Making non-persistent CGAs persistent to make the non-persistent CGA's persistent.

## 2.1.8.6 IPv6 provider edge over MPLS (6PE)

6PE allows IPv6 domains to communicate with each other over an IPv4 MPLS core network. Because forwarding is based on MPLS labels, backbone infrastructure upgrades and core router re-configuration is not required in this architecture. 6PE is a cost-effective solution for IPv6 deployment.

*Figure 9: Example of a 6PE topology within one AS*




### 2.1.8.6.1 6PE control plane support

The 6PE MP-BGP routers support:

- IPv4 and IPv6 dual-stack
- MP-BGP to exchange IPv6 reachability information:
  - The 6PE routers exchange IPv6 reachability information using MP-BGP (AFI 2, SAFI 4).

  - An IPv4 address of the 6PE router is encoded as an IPv4-mapped IPv6 address in the BGP next-hop field. This is usually the IPv4 system address.
  - The 6PE router binds MPLS labels to the IPv6 prefixes it advertises. SR OS routers advertise the IPv6 explicit null (value 2) in advertised 6PE routes but accept any arbitrary label from peers.
- The most preferred tunnel to the BGP next-hop allowed by the 6PE resolution filter is used to tunnel the traffic to the remote 6PE router. Use the following command to configure the 6PE resolution filter.

```
configure router bgp next-hop-resolution labeled-routes transport-tunnel family resolution-filter
```

#### 2.1.8.6.2 6PE data plane support

The ingress 6PE router can push two or more MPLS labels to send the packets to the egress 6PE router. The top labels are associated with resolving the transport tunnels. The bottom label is advertised in MP-BGP by the remote 6PE router. Typically, the IPv6 explicit null (value 2) label is used, but any arbitrary value can be received when the remote 6PE router is not an SR OS router.

The egress 6PE router pops the top transport labels. When the IPv6 explicit null label is exposed, the egress 6PE router knows that an IPv6 packet is encapsulated. It pops the IPv6 explicit null label and performs an IPv6 route lookup to find the next hop for the IPv6 packet.

### 2.1.9 Static route resolution using tunnels

Use the commands in the following context to forward packets of a static route to an indirect next-hop over a tunnel programmed in TTM:

- **MD-CLI**

```
configure router static-routes route indirect tunnel-next-hop
```

  In the MD-CLI, if the **tunnel-next-hop** context is configured and **resolution** is set to **none**, the binding to the tunnel is removed and resolution resumes in RTM to IP next-hops.

- **classic CLI**

```
configure router static-route-entry indirect tunnel-next-hop
```

  In the classic CLI, if the **tunnel-next-hop** context is configured and **resolution** is set to **disabled**, the binding to the tunnel is removed and resolution resumes in RTM to IP next-hops.

If the **resolution** is set to **any**, any supported tunnel type in the static route context is selected following TTM preference.

The following tunnel types are supported in a static route context: LDP, RSVP-TE, Segment Routing (SR) Shortest Path, and Segment Routing Traffic Engineering (SR-TE):

- **LDP**

  The **ldp** command option instructs the code to search for an LDP LSP with a FEC prefix corresponding to the address of the indirect next-hop. Both LDP IPv4 FEC and LDP IPv6 FEC can be used as the tunnel next-hop. However, only an indirect next-hop of the same family (IPv4 or IPv6) as the prefix of

the route can use an LDP FEC as the tunnel next-hop. In other words, an IPv4 (IPv6) prefix can only be resolved to an LDP IPv4 (IPv6) FEC.

- **RSVP-TE**

  The **rsvp-te** command option instructs the code to search for the set of lowest metric RSVP-TE LSPs to the address of the indirect next-hop. The LSP metric is provided by MPLS in the tunnel table. The static route treats a set of RSVP-TE LSPs with the same lowest metric as an ECMP set.

  The user has the option of configuring a list of RSVP-TE LSP names to be used exclusively instead of searching in the tunnel table. In that case, all LSPs must have the same LSP metric in order for the static route to use them as an ECMP set. Otherwise, only the LSPs with the lowest common metric value are selected.

  A P2P auto-lsp that is instantiated via an LSP template can be selected in TTM when **resolution** is set to **any**. However, it is not recommended to configure an **auto-lsp** *name* explicitly under the **rsvp-te** node as the auto-generated name can change if the node reboots, which blackholes the traffic of the static route.

- **SR shortest path**

  When the **sr-isis** or **sr-ospf** command options are enabled, an SR tunnel to the indirect next-hop is selected in the TTM from the lowest preference IS-IS or OSPF instance, and if many instances have the same lowest preference, it is selected from the lowest numbered IS-IS or OSPF instance. Both SR-ISIS IPv4 and SR-ISIS IPv6 tunnels can be used as tunnel next-hops. However, only an indirect next-hop of the same family (IPv4 or IPv6) as the prefix of the route can use an SR-ISIS tunnel as a tunnel next-hop. In other words, an IPv4 (IPv6) prefix can only be resolved to a SR-ISIS IPv4 (IPv6).

- **SR-TE**

  The **sr-te** command option instructs the code to search for the set of lowest metric SR-TE LSPs to the address of the indirect next-hop. The LSP metric is provided by MPLS in the tunnel table. The static route treats a set of SR-TE LSPs with the same lowest metric as an ECMP set.

  The user has the option of configuring a list of SR-TE LSP names to be used exclusively instead of searching in the tunnel table. In that case, all LSPs must have the same LSP metric in order for the static route to use them as an ECMP set. Otherwise, only the LSPs with the lowest common metric value are selected.

Realize that the resolution filter, under static-route entry, does not validate the provided lsp-name type of the LSP against the requested filter context protocol type.

If one or more explicit tunnel types are specified using the **resolution-filter** command option, only these tunnel types are selected again following the TTM preference.

The user must set **resolution** to **filter** to activate the list of tunnel-types configured under resolution-filter.

If **disallow-igp** is enabled, the static route is not activated using IP next-hops in RTM if no tunnel next-hops are found in TTM.

### 2.1.9.1 Static route ECMP support

The following is the ECMP behavior of a static route:

- ECMP is supported when resolving in RTM multiple static routes of the same prefix with multiple user-entered indirect IP next-hops. The system picks as many direct next-hops as available in RTM beginning from the first indirect next-hop and up to the value of the **ecmp** command option in the system.

- ECMP is also supported when resolving in TTM a static route to a single indirect next-hop using a LDP tunnel when LDP has multiple direct next-hops.
- ECMP is supported when resolving in TTM a static route to a single indirect next-hop using a RSVP-TE tunnel type when there is more than one RSVP LSP with the same lowest metric to the indirect next-hop.
- ECMP is supported when resolving in TTM a static route to a single indirect next-hop using a list of user-configured RSVP-TE LSP names when these LSPs have the same metric to the indirect next-hop.
- ECMP is supported when resolving in TTM multiple static routes of the same prefix with multiple user-entered indirect next-hops, each binding to a tunnel type. The system picks as many tunnel next-hops as available in TTM beginning from the first indirect next-hop and up to the value of the **ecmp** command option in the system. The spraying of flow packets is performed over the entire set of resolved next-hops that correspond to the selected indirect next-hops.
- ECMP is supported when resolving concurrently in RTM and TTM multiple static routes of the same prefix with multiple user-entered indirect tunnel next-hops. There is no support for mixing IP and tunnel next-hops for the same prefix using different indirect next-hops. Tunnel next-hops are preferred over IP next-hops.

### 2.1.9.2 Static route using flexible algorithms tunnels

When configuring a static route toward an indirect next hop, the path selection based upon the constraints of a particular Flex-Algorithm should be considered. In such a use case, it is necessary to steer traffic into a corresponding flexible algorithm segment routing tunnel. This can be achieved with the **tunnel-next-hop flex-algo** command. This uses the specified flexible algorithm to construct a tunnel toward the indirect static-route next-hop.

The use of this command assumes that the router is participating in the flexible algorithm. This command instructs the router to lookup the indirect next-hop using flexible algorithm tunnels. The static route is not activated if a flexible algorithm-aware tunnel does not exist in the indirect next-hop.

When a router receives an IP packet, the static-route entry may steer toward the indirect next-hop using a flexible algorithm-aware SR tunnel, provided that such a tunnel exists. If the tunnel does not exist, the route is not active and the received IP packet is dropped, as long as no longest prefix match (LPM) route exists.

When the **flex-algo** command is configured, the resolution filter can only use matching flexible algorithm-aware SR tunnels created by flex-algo aware routing protocols (for example, SR IS-IS). If such an entry does not exist in the tunnel-table, the static-route entry does not become active.

Use the commands in the following context to configure static routes using flexible algorithms:

- **MD-CLI**

```
configure router static-routes route indirect tunnel-next-hop flex-algo
```

- **classic CLI**

```
configure router static-route-entry indirect tunnel-next-hop
```

## 2.2 Weighted load balancing over MPLS LSP

The weighted load-balanced, or weighted-ecmp, feature sprays packets of IGP, BGP, and static route prefixes, resolved to a set of ECMP tunnel next hops, proportionally to the weights configured for each MPLS LSP in the ECMP set.

Weighted load balancing is supported in the following forwarding contexts:

- IGP prefix resolved to IGP shortcuts in RTM (**igp-shortcut** or **advertise-tunnel-link** enabled in the IGP instance)
- BGP prefix with the BGP next hop resolved to IGP shortcuts in RTM
- Static route prefix resolved to an indirect next hop, which is resolved to a set of equal-metric MPLS LSPs in TTM. The user can allow automatic selection or specify the names of the equal-metric MPLS LSPs in TTM to be used in the ECMP set.
- Static route prefix resolved to an indirect next hop, which is resolved to IGP shortcuts in RTM
- BGP prefix with a BGP next hop resolved to a static route, which resolves to a set of tunnel next hops toward an indirect next hop in RTM or TTM
- BGP prefix resolving to another BGP prefix, whose next hop is resolved to a set of ECMP tunnel next hops with a static route in RTM or TTM or to IGP shortcuts in RTM
- IPv4 BGP-labeled unicast routes whose next hop resolves to a set of tunnels in TTM
- BGP-labeled IPv6 packets (6PE) resolving in TTM

This feature does not modify the route calculation: the same set of ECMP next hops is computed for a prefix. The feature also does not change the hash routine; only the spraying of the flows over the tunnel next hops is modified to reflect the normalized weight of each tunnel next hop.

Static route implementation supports ECMP over a set of equal-cost MPLS LSPs. The user can allow automatic selection or specify the names of the equal-metric MPLS LSPs in TTM to be used in the ECMP set. For more information, see Static route resolution using tunnels.

Weighted load balancing can be applied whether a route resolves directly over an ECMP set of RSVP or SR-TE LSPs with a configured load balancing weight, or where resolution occurs via a BGP tunnel which in turn uses an ECMP set of RSVP or SR-TE LSPs with a configured load balancing weight.

### 2.2.1 Weighted load balancing IGP, BGP, and static route prefix packets over IGP shortcut

#### 2.2.1.1 Feature configuration

The user must have the IGP shortcut or forwarding adjacency feature enabled in one or more IGP instances. Use the commands in the following contexts to configure the IGP shortcut or forwarding adjacency features:

- **IS-IS**

```
configure router isis igp-shortcut
configure router isis advertise-tunnel-link
```

- **OSPF**

```
configure router ospf igp-shortcut
configure router ospf advertise-tunnel-link
```

Use the following command to disable specific MPLS LSPs from being used in IGP shortcut or forwarding adjacency:

- **MD-CLI**

```
configure router mpls lsp igp-shortcut admin-state disable
```

- **classic CLI**

```
configure router mpls lsp no igp-shortcut
```

Use the following command to enable the weighted load balancing feature.

```
configure router weighted-ecmp
```

When this command is enabled, packets of IGP, BGP, and static route prefixes resolved to a set of ECMP tunnel next-hops are sprayed proportionally to the weights configured for each MPLS LSP in the ECMP set.

Use the following command to configure a weight for each LSP.

```
configure router mpls lsp load-balancing-weight weight
```

Use the following command to configure the weight for an auto-LSP signaled via an LSP template.

```
configure router mpls lsp-template load-balancing-weight weight
```

There is no default weight value for an LSP. If any LSP in the ECMP set of a prefix does not have a weight configured, the regular ECMP spraying for the prefix is performed. The user-entered weight is normalized to the closest integer value that represents the number of entries in the ingress prefix hash table assigned to the LSP for the purpose of spraying packets of all prefixes resolved to this LSP. The higher the normalized weight, the more entries are assigned to the LSP, and the more packets are sent to this LSP.

### 2.2.1.2 Feature behavior

This section describes the behavior of the weighted load-balancing feature for IGP, BGP, and static route prefixes resolved in RTM to IGP shortcuts.

When an IGP, BGP, or a static route prefix is resolved in RTM to a set of ECMP tunnel next-hops of type RSVP-TE, and the router level **weighted-ecmp** command option is enabled, the ingress hash table for the next-hop selection is populated with a number of tunnel next-hop entries for each LSP equal to the normalized LSP weight value. All prefixes resolving to the same set of ECMP tunnel next-hops use the same table.

This feature performs the following:

1. MPLS populates the user-configured LSP weight in TTM. When the **weighted-ecmp** global command is enabled, and any LSP in the ECMP set of a prefix does not have a weight configured, the regular ECMP spraying for the prefix is performed.

2. IGP computes the normalized weight for each prefix tunnel next-hop. The minimum value of the normalized weight is 1 and the maximum is 64. IGP updates the route in RTM with the set of tunnel next-hops and normalized weights. RTM downloads the information to IOM for inclusion in the FIB.
3. The normalized weights of route tunnel next-hops are updated in the following cases:
   - When the main SPF is run following a trigger, for example, network failure, and updates a route with a modified set of tunnel next-hops. This triggers a route re-download to the IOM and all users of RTM are notified.
   - The user adds or changes the weight of one or more LSPs. In this case, RTM performs a route download to IOM, but other users of RTM are not notified because the route resolution did not change.
4. The weighted load balancing feature is only applied to a prefix when all the tunnel next-hops in the ECMP set have the same endpoint. If an IGP prefix resolves in RTM to a set of ECMP tunnel next-hops that do not terminate on the same endpoint, the regular ECMP spraying is performed. If BGP performs BGP ECMP to a set of BGP ECMP next-hops for a prefix (weighted-bgp-ecmp-prd), regular ECMP spraying is performed toward a BGP next-hop if the subset of its tunnel next-hops does not terminate on the same endpoint.
5. Regular ECMP spraying is also applied if a prefix is resolved in RTM to an ECMP set that consists of a mix of IP and tunnel next-hops.
6. This feature is not supported in the following contexts:
   - Packets of BGP prefix with the BGP next-hop resolved in TTM to RSVP LSP (BGP shortcut).
   - CPM generated packets, including OAM packets, which are looked-up in RTM and which are forwarded over tunnel next-hops. These are forwarded using either regular ECMP or by selecting one next-hop from the set.

#### 2.2.1.3 ECMP considerations

The weight assigned to an LSP affects only the forwarding decision, not the routing decision. It does not change the selection of the set of ECMP tunnel next-hops of a prefix when more next-hops exist than the value of the router **ecmp** command option. This selection continues to follow the algorithm used in the IGP shortcut feature.

After the set of tunnel next-hops is selected, the LSP weight is used to modulate the amount of packets forwarded over each next-hop.

#### 2.2.1.4 Weighted load balancing static route packets over MPLS LSP

##### 2.2.1.4.1 Feature configuration

The configuration of the resolution of a static route prefix to set of MPLS LSPs is described in Static route resolution using tunnels which also provides the selection rules among multiple LSP types: RSVP-TE, SR-TE, LDP, SR-ISIS, and SR-OSPF. A static route of a prefix can only be resolved to a set of tunnel next-hops of the same type though, for each indirect next-hop.

To perform ECMP over a set of configured MPLS LSPs, the user must enter two or more LSP names to be used as tunnel next-hops. If automatic selection is performed, ECMP is performed if two or more MPLS

LSPs are in TTM to the indirect next-hop of the static route. However, all LSPs must have the same LSP metric; otherwise, only the tunnel next-hops with the same lowest metric are activated for the static route.

Use the following command to force the metric of an LSP to a constant value.

```
configure router mpls lsp metric
```

If the user enters, for the same static route, more LSP names with the same LSP metric than the value of the router level **ecmp** command option, only the first configured LSPs equal to the **ecmp** value are selected. The remaining tunnel next-hops for the route are not activated. When automatic MPLS LSP selection is performed in TTM, the lowest tunnel ID is used as a tie-breaker among the same lowest metric LSPs.

To perform weighted load-balancing over the set of MPLS LSPs, either when the LSP names are provided or when auto-selection in TTM is performed, the user must also enable the weighted ECMP globally like for static, IGP, and BGP prefixes resolving to IGP shortcuts.

Use the following command to enable weighted ECMP globally.

```
configure router weighted-ecmp
```

#### 2.2.1.4.2 Feature behavior

The behavior of this feature in terms of RTM and IOM is exactly the same as in the case of BGP, IGP, and static route prefixes resolving to IGP shortcuts. See Feature behavior for more information. In this case, the static route module computes the normalized weight for each prefix tunnel next-hop of the static route indirect next-hop. The minimum value of the normalized weight is 1 and the maximum is 64. The static route module updates the route in RTM with the set of tunnel next-hops and normalized weights. RTM downloads the information to IOM for inclusion in the FIB.

If any LSP in the ECMP set of a prefix static route does not have a weight configured, the regular ECMP spraying for the prefix is performed.

ECMP is also supported when resolving in TTM the same static route with multiple user-entered indirect next-hops, each binding to the same or different tunnel types. The system picks as many tunnel next-hops as available in RTM, beginning from the first indirect next-hop and up to the value of the **ecmp** command option in the system. In this case, the weighted load-balancing is applied directly using the weights of the selected set of tunnel next-hops. If any LSP in the ECMP set of a prefix static route does not have a weight configured, or if any of the indirect next-hops binds to an LDP LSP, the regular ECMP spraying for the prefix is performed.

If the same prefix is resolved via both a static route and an IGP shortcut route, the RTM default protocol preference installs the static route only. Therefore, the set of ECMP tunnel next-hops and the weighted load balancing behavior are determined by the static route configuration and not by the IGP shortcut configuration.

### 2.2.2 Weighted load balancing for 6PE and BGP IPv4-labeled unicast routes

Use the following command to control the ECMP-like spraying for BGP-labeled IPv6 packets (6PE) and BGP-labeled IPv4 unicast routes resolving to tunnels in TTM, where the maximum number of ECMP router

represents the maximum number of RSVP and SR-TE tunnels in the set representing equal-cost paths to the BGP next hop.

```
configure router ecmp
```

Use the following command to configure weighted ECMP behavior, where the load-balancing weight of the tunnel is considered in the packet spraying behavior.

```
configure router bgp next-hop-resolution weighted-ecmp
```

Weighted ECMP is disabled by default.

### 2.2.3 Strict weighted load-balancing

Strict weighted load-balancing is enabled by configuring **weighted-ecmp strict** in global routing mode. The strict enforcement for a load balancing weight is valid for both a base router instance and for a VPRN instance.

- With strict enforcement, a weight must be configured on each interface within a wECMP interface bundle before the interface is taken into wECMP operation.
- Without **weighted-ecmp strict** enforcement enabled, and if one or more interfaces within a wECMP interface bundle does not have a **load-balancing-weight** *weight* configured, then the wECMP load-balancing falls back to classic ECMP operation and equally sprays data-plane traffic across the available interfaces.
- A special case of **weighted-ecmp strict** is when none of the available paths or next-hops have a **load-balancing-weight** value associated. Then, the load-balancing falls back to the classic ECMP.
- **weighted-ecmp strict** is enabled in the router global context for IS-IS, OSPF, and OSPFv3. Other routing technologies follow classic **weighted-ecmp** operation.

## 2.3 Class-based forwarding of IPv4/IPv6 prefix over IGP IPv4 shortcut

This feature enables class-based forwarding (CBF) over IGP shortcuts. When the **class-forwarding** command is enabled, the following types of packets are forwarded based on their forwarding class:

- packets of BGP prefixes
- packets that are CPM-originated for the IPv4, IPv6, or both IPv4 and IPv6 families that have been enabled over IGP shortcuts using the **igp-shortcut** context in one or more IGP instances

The SR OS CBF implementation supports spraying of packets over a maximum of six forwarding sets of ECMP LSPs. The user must define a class-forwarding policy object in MPLS to configure the mapping of FCs to the forwarding sets. Then, the user assigns the CBF policy name and set ID to each MPLS LSP that is used in IGP shortcuts.

When a BGP IPv4 or IPv6 prefix is resolved, the FC of the packet, is used to look up the forwarding set ID. Then, a modulo operation is performed on the tunnel next-hops of this set ID only, to spray packets of this FC. The datapath concurrently implements, CBF and ECMP within the tunnels of each set ID.

CPM-originated packets on the router, including control plane and OAM packets, are forwarded over a single LSP from the set of LSPs that the packet's FC is mapped to, as per the CBF configuration.

### 2.3.1 Feature configuration

Use the following command to configure CBF over IGP shortcuts.

```
configure router mpls class-forwarding-policy
```

All FCs are mapped to set 1 as soon as the policy is created. The user can make changes to the mapping of FCs as required. An FC, which is not added to the class-forwarding policy, is therefore always mapped to set 1. At most, an FC can be mapped to a single forwarding set. One or more FCs can map to the same set. Use the following command to configure the FC to a set other than set 1.

```
configure router class-forwarding-policy fc forwarding-set
```

The user can indicate the initial default set by including the **default-set** command option. Use the following command to configure the **default-set** command option.

```
configure router mpls class-forwarding-policy default-set
```

The default forwarding set is used to forward packets of any FC in cases where all LSPs of the forwarding set the FC maps to become operationally down. The router uses the user-configured default set as the initial default set. Otherwise, the router elects the lowest numbered set as the default forwarding set in a class-forwarding policy. When the last LSP in a default forwarding set goes into an operationally down state, the router designates the next lowest-numbered set as the new default forwarding set.

A mapping to a class-forwarding policy and set is added to the existing CBF configuration of an RSVP-TE or SR-TE LSP or to an LSP template. Use the following commands to perform this mapping function.

```
configure router mpls lsp class-forwarding forwarding-set policy set
configure router mpls lsp-template class-forwarding forwarding-set policy set
```

An MPLS LSP can map only to a single class-forwarding policy and forwarding set. Multiple LSPs can map to the same policy and set. If they form an ECMP set, from the IGP shortcut perspective, packets of the FCs mapped to this set are sprayed over these LSPs based on a modulo operation of the output of the hash routine on the packet's headers and the number of LSPs in the set.

### 2.3.2 Feature behavior

When a BGP IPv4 or IPv6 prefix is resolved to a BGP next-hop, consisting of up to 64 resolved next-hops (LSPs and IP links), the default behavior of the datapath is to spray the packets over the entire ECMP set using a modulo operation of the number of resolved next-hops in the ECMP set and the output of the hash on the packet header fields.

Both the CBF feature in LDP-over-IGPv4 shortcuts and this CBF feature over IGP IPv4 shortcuts make use of the CBF class-forwarding policy. IGP always passes the CBF information populated by MPLS for each LSP used as a tunnel next-hop by an IGP prefix. The new CBF information is checked for consistency. If more than a single class-forwarding policy exists in the tunnel next-hops of a IGP prefix, IGP removes the new CBF information from all the corresponding tunnels and the behavior is as if there were no CBF info.

Use the following command to enable the CBF feature.

```
configure router class-forwarding
```

When the CBF feature is enabled, each application (BGP, CPM), when looking up a prefix in RTM, finds up to 64 IP and tunnel next-hops. This lookup is split in two subsets:

- Subset 1— tunnel next-hops with new CBF information (FCs mapped to this LSP, default LSP (true/false), CBF Policy ID>0, Set ID>0). This information is usable by both LDP and other applications.
- Subset 2— tunnel-next-hops with no CBF information and IP next-hops. Usable by all applications, except that LDP uses tunnel next-hops only.

The BGP application performs a lookup in RTM for a prefix matching each BGP next-hop of a prefix. The BGP application selects tunnels belonging to the class-forwarding sets in Subset 1 and for each BGP next-hop of a prefix. The remaining tunnels, with no CBF configuration and the IP next-hops, are still programmed to IOM. However, BGP and the datapath uses them only when all the class-forwarding sets are not available as described in the information that follows.

SR OS implements a hierarchical ECMP architecture for BGP prefixes in the datapath. The first level is the ECMP at the BGP next-hop level. The second level is ECMP at the resolved next-hop (IP or tunnel next-hop) level. The CBF feature is independently applied to the set of resolved tunnel next-hops of each BGP next-hop of a prefix. The user must make sure that the sets of LSPs that are used as IGP shortcuts to reach each of the BGP next-hops have the appropriate FC mappings.

The following procedures are enforced in the CBF feature:

- The tunnels in the full next-hop ECMP set, with set size greater or equal to 1 and less than or equal to 64, can use MPLS LSPs that terminate on multiple endpoints (BGP next-hop itself or otherwise) to reach the next-hop of a BGP prefix. The existing ECMP tunnel and IP next-hop selection behavior, when resolving a prefix over IGP shortcuts, continues to be used.
- If no LSP among the full ECMP set of a BGP next-hop has a class-forwarding policy configuration assigned, then the set is considered inconsistent from a CBF perspective. No CBF-related information is programmed in IOM and regular ECMP spraying over the full set occurs.
- If only a single class-forwarding policy is referenced by one or more LSPs in the full ECMP set of a BGP next-hop, the full set is considered consistent from a CBF perspective, and the class-forwarding policy is used to spray packets of each FC over the LSPs within each forwarding set. As a result of this processing, only the LSPs that have been selected for forwarding traffic are programmed in IOM with CBF information. The remaining LSPs and IP next-hops of the BGP next-hop, are also programmed in IOM but without any CBF information associated and, therefore, is not used for CBF.
- If multiple class-forwarding policies are referenced by LSPs in the full ECMP set of a BGP next-hop, the set is considered inconsistent from a CBF perspective. No CBF related information is programmed in IOM and regular ECMP spraying over the full set occurs.

The following describes the fallback behavior in datapath of the CBF feature:

- An FC, for which all LSPs in the forwarding set are operationally down, has its packets forwarded over the default forwarding set. The default forwarding set is either the initial default forwarding set configured by the user or the lowest numbered set in the class-forwarding policy that has one or more LSPs in the operationally UP state. If the initial or subsequently elected default forwarding set has all its LSPs operationally down, the next lower numbered forwarding set, which has at least one LSP in the operationally up state, is elected as the default forwarding set.
- If all LSPs of all forwarding sets become operationally down, the router resumes regular ECMP spraying on the remaining LSPs and IP next-hops in the full ECMP set.
- Whenever the first LSP in a forwarding set becomes operationally UP, the router triggers the re-election of the default set and selects this set as the new default set, if it is the initial default set, otherwise, it selects lowest numbered set.

### 2.3.3 Feature limitations

The following are the limitations of the CBF feature.

- CBF applies to packets of IPv4 and IPv6 BGP prefixes only. CBF does not apply to IGP prefixes and static route prefixes resolved over IGP IPv4 shortcuts. The latter are forwarded using regular ECMP over the entire set of up to 64 tunnel next-hops.
- CPM originated packets on the router, including the control plane and OAM packets, are forwarded over a single LSP from the set of LSPs the packet's FC is mapped to, as per the CBF configuration. The CPM, however, only maintains a maximum of 64 next-hops for a specified destination prefix. Therefore, if there are multiple BGP next-hops for a prefix, CPM selects 64 tunnel next-hops by cycling over the BGP next-hops in ascending order. Then, the first LSP in the first set ID that the FC of the packet maps to is selected to forward the packet.

  Furthermore, the CBF information consistency check, the CBF default set determination, and the CBF set failover procedures are applied to this set of 64 tunnel next-hops.

  The user can configure the SGT-QoS feature to change the DSCP and FC of CPM-originated packets of a specific control plane protocol to select an LSP from a different set ID. This configuration allows, for instance, the forwarding of BGP Keep-Alive packets over an LSP of the same set ID as that of the data plane packets of the BGP prefixes destined for the same BGP next-hop.
- Weighted ECMP, at the transport tunnel level of BGP prefixes over IGP shortcuts, and the CBF feature on a per-BGP next-hop basis are mutually exclusive. Specifically, if the user enables both weighted ECMP and CBF, weighted ECMP applies as long as all the LSPs used as tunnel next-hops to reach the BGP next-hop of a prefix have a user-configured weight. Otherwise, the CBF feature applies as per the procedures described in Feature behavior.

  Use the following commands to configure weighted ECMP and class-forwarding respectively.

```
configure router weighted-ecmp
configure router class-forwarding
```

### 2.3.4 Datapath support

When a packet of a BGP IPv4 or IPv6 prefix is received, the datapath uses the FC that the packet was classified into to look up the forwarding set ID. The datapath then performs a modulo operation on the tunnel next-hops of this set ID, to select the one next-hop for forwarding the packet. Therefore, packets matching an FC are only sprayed over the ECMP tunnel-next-hops of the set ID this FC maps to.

Both the BGP or CPM application and IOM use the same algorithm for failover and default class-forwarding set determination, as described in Feature behavior and illustrated in Example configuration and default CBF set election.

If MPLS deletes an LSP from a specified set ID, the IOM handles failover within the same set ID. The IOM reprograms the datapath to spray packets of the impacted FCs over the remaining tunnel next-hops of the set ID.

Similarly, the IOM handles failover between class-forwarding sets when MPLS deletes the last LSP in a set ID. The IOM reprograms the datapath to spray packets of the impacted FCs over the tunnel next-hops of the failover set ID. In both cases, the failover does not make use of the uniform failover procedure; however, if an LSP activated its FRR backup path, it remains in the set ID and continues to forward traffic of the mapped FCs.

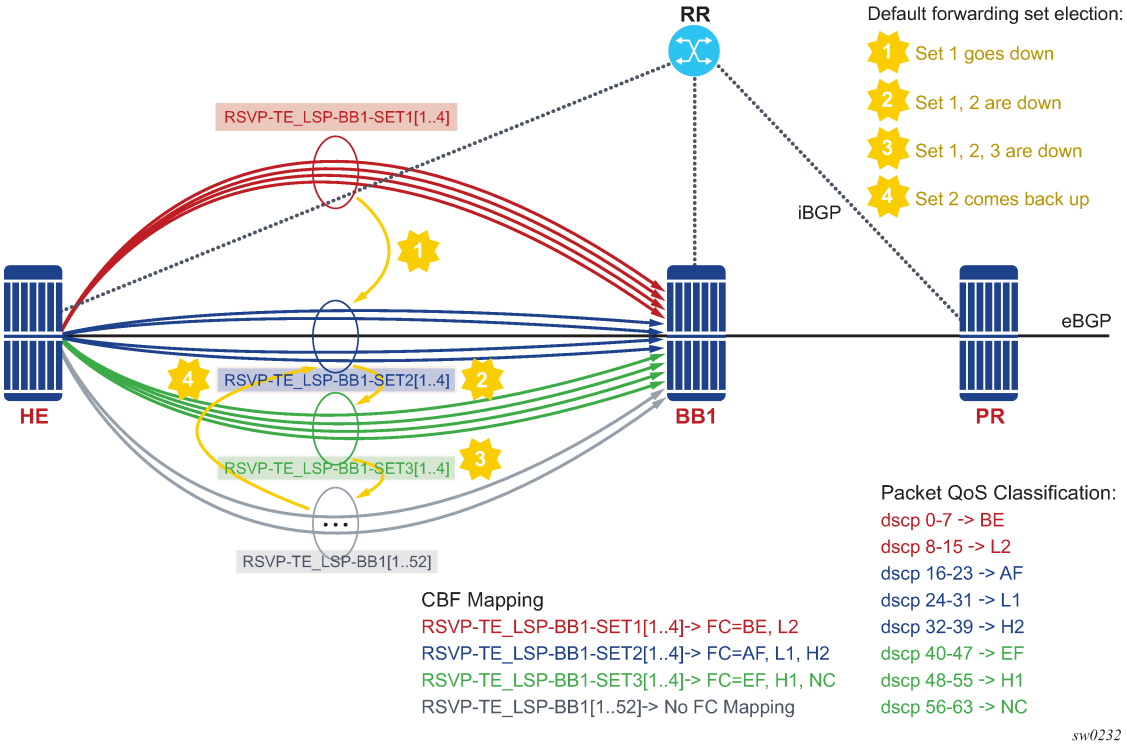
Finally, BGP updates the set IDs, used to reach a BGP next-hop, any time IGP updates the information in the RTM.

### 2.3.5 Example configuration and default CBF set election

Assume the following user configuration:

- The FC mapping to the sets and the default forwarding set election are illustrated in Figure 10: Default forwarding set election.
- All sets and RSVP-TE LSPs outside of the three class-forwarding sets are up initially.
- Set 1 is elected as the default class-forwarding set (because the user did not configure an initial default set).
- If All LSPs in Set 1 go operationally down, Set 2 is elected as the default class-forwarding set.
- If Set 2 subsequently goes down, Set 3 is elected as the default class-forwarding set.
- If Set 3 subsequently goes down, then packets of BGP prefixes are ECMP sprayed over the remaining non-CBF RSVP-TE LSPs.
- If Set 2 comes back up, then Set 2 is elected as the default class-forwarding set.

*Figure 10: Default forwarding set election*




The following example shows class-forwarding as true, which enables the CBF feature for BGP and CPM traffic. IGP shortcut is enabled in this IS-IS instance with both families IPv4 and IPv6 resolving to RSVP-TE LSPs. Four LSPs exist in set 1, set 2, and set 3. The final LSP configuration has no CBF options for a total of 64 LSPs to BB1.

### Example: Class-based forwarding of IPv4 and IPv6 prefixes over IGP IPv4 shortcut (MD-CLI)

```
[ex:/configure router "Base"]
A:admin@node-2# info
    class-forwarding true
    interface "system" {
        ipv4 {
            primary {
                address 192.0.2.194
                prefix-length 32
            }
        }
        ipv6 {
            address 3ffe::a14:194 {
                prefix-length 128
             }
        }
    }
    interface "toSim199" {
        port 1/1/1
        ipv4 {
            primary {
                address 10.202.5.194
                prefix-length 24
            }
            secondary 10.101.0.194 {
                prefix-length 32
            }
        }
        ipv6 {
            address 2001:db8:a0b:12f0::1 {
                prefix-length 64
             }
        }
    }
    interface "toSim213" {
        port 1/1/2
        ipv4 {
            primary {
                address 10.202.4.194
                prefix-length 24
            }
        }
    }
    interface "toSim219" {
        port 1/1/3
        ipv4 {
            primary {
                address 10.202.8.194
                prefix-length 24
            }
        }
    }
[ex:/configure router "Base"]
A:admin@node-2# info
...
    isis 0 {
       igp-shortcut {
            tunnel-next-hop {
                family ipv4 {
                    resolution filter
                    resolution-filter {
```

```
                        rsvp true
                    }
                }
                family ipv6 {
                    resolution filter
                    resolution-filter {
                        rsvp true
                    }
                }
            }
        }
    }
[ex:/configure router "Base" mpls]
A:admin@node-2# info
    cspf-on-loose-hop true
    interface "system" {
    }
    interface "toSim199" {
    }
    interface "toSim213" {
        admin-group ["olive"]
    }
    interface "toSim219" {
    }
    class-forwarding-policy "cbf1" {
        fc be {
            forwarding-set 1
        }
        fc l2 {
            forwarding-set 1
        }
        fc af {
            forwarding-set 2
        }
        fc l1 {
            forwarding-set 2
        }
        fc h2 {
            forwarding-set 2
        }
        fc ef {
            forwarding-set 3
        }
        fc h1 {
            forwarding-set 3
        }
        fc nc {
            forwarding-set 3
        }
    }
    path "empty" {
    }
    lsp "RSVP-TE_LSP-BB1-SET1" {
        type p2p-rsvp
        to 192.1.2.194
        class-forwarding {
            forwarding-set {
                policy "cbf1"
                set 1
            }
        }
        primary "empty" {
        }
    }
```

```
    lsp "RSVP-TE_LSP-BB1-SET2" {
        type p2p-rsvp
        to 192.1.2.194
        class-forwarding {
            forwarding-set {
                policy "cbf1"
                set 2
            }
        }
        primary "empty" {
        }
    }
    lsp "RSVP-TE_LSP-BB1-SET3" {
        type p2p-rsvp
        to 192.1.2.194
        class-forwarding {
            forwarding-set {
                policy "cbf1"
                set 3
            }
        }
        primary "empty" {
        }
    }
    lsp "RSVP-TE_LSP-B1[1..52]" {
        type p2p-rsvp
        to 192.1.2.194
        primary "empty" {
        }
    }
```

### Example: Class-based forwarding of IPv4 and IPv6 prefixes over IGP IPv4 shortcut (classic CLI)

```
A:node-2>config>router# info
#--------------------------------------------------
echo "IP Configuration"
#--------------------------------------------------
        interface "system"
            address 192.0.2.194/32
            ipv6
                address 3ffe::a14:194/128
            exit
        exit
        interface "toSim199"
            address 10.202.5.194/24
            secondary 10.101.0.194/32
            port 1/1/1
            ipv6
                address 2001:db8:a0b:12f0::1/64
            exit
        exit
        interface "toSim213"
            address 10.202.4.194/24
            port 1/1/2
        exit
        interface "toSim219"
            address 10.202.8.194/24
            port 1/1/3
        exit
        class-forwarding
   // Enables CBF feature for BGP and CPM traffic
```

```
A:node-2>config>router>isis# info
----------------------------------------------
            igp-shortcut
//  Enables IGP shortcut in this ISIS instance with both families IPv4 and IPv6
resolving to RSVP-TE LSPs
                tunnel-next-hop
                    family ipv4
                        resolution filter
                        resolution-filter
                            rsvp
                        exit
                    exit
                    family ipv6
                        resolution filter
                        resolution-filter
                            rsvp
                        exit
                    exit
                exit
            exit
----------------------------------------------
A:node-2>config>router>mpls# info
----------------------------------------------
            class-forwarding-policy cbf1
                fc be forwarding-set 1
                fc l2 forwarding-set 1
                fc af forwarding-set 2
                fc l1 forwarding-set 2
                fc h2 forwarding-set 2
                fc ef forwarding-set 3
                fc h1 forwarding-set 3
                fc nc forwarding-set 3
            cspf-on-loose-hop
            exit
            interface "system"
            exit
            interface "toSim199"
            exit
            interface "toSim213"
                admin-group "olive"
            exit
            interface "toSim219"
            exit
            path "empty"
            exit
            lsp "RSVP-TE_LSP-BB1-SET1[1..4]"  // Four LSPs in Set1
                to 192.0.2.194/32
                cspf
                class-forwarding
                    forwarding-set policy “cbf1” set 1
                exit
                primary "empty"
                exit
            exit
            lsp "RSVP-TE_LSP-BB1-SET2[1..4]"  // Four LSPs in Set2
                to 192.0.2.194/32
                cspf
                class-forwarding
                    forwarding-set policy “cbf1” set 2
                exit
                primary "empty"
                exit
            exit
            lsp "RSVP-TE_LSP-BB1-SET3[1..4]"  // Four LSPs in Set3
```

```
                to 192.0.2.194/32
                cspf
                class-forwarding
                    forwarding-set policy “cbf1” set 3
                exit
                primary "empty"
                exit
            exit
            lsp "RSVP-TE_LSP-BB1[1..52]"  //
 Other LSP configuration with no CBF options for a total of 64 LSPs to BB1
                to 192.0.2.194/32
                cspf
                primary "empty"
                exit
            exit
-------------------------------------------
```

## 2.4 Aggregate next hop

This feature adds the ability to configure an indirect next-hop for aggregate routes. The indirect next-hop specifies where packets are forwarded if they match the aggregate route, but is not a more-specific route in the IP forwarding table.

## 2.5 Invalidate next-hop based on ARP/neighbor cache state

This feature invalidates next-hop entries for static routes when the next-hop is no longer reachable on directly connected interfaces. This invalidation is based on ARP and Neighbor Cache state information.

When a next-hop is detected as no longer reachable because of ARP/neighbor cache expiry, the route’s next-hop is set as unreachable to prevent the SR from sending continuous ARPs/neighbor solicitations triggered by traffic destined for the static route prefix. When the next-hop is detected as reachable via ARP or neighbor advertisements, the state of the next-hop is set back to valid.

### 2.5.1 Invalidate next-hop based on IPv4 ARP

This feature invalidates a static route based on the reachability of the next-hop in the ARP cache when the **validate-next-hop** command is enabled for an IPv4 static route. Use the commands in the following contexts to enable the validate-next-hop feature for an IPv4 static route:

- **MD-CLI**

```
configure router static-routes route next-hop
configure service vprn static-routes route next-hop
```

- **classic CLI**

```
configure router static-route-entry next-hop
configure service vprn static-route-entry next-hop
```

In this case, when the ARP entry for the next-hop is INVALID or not populated, the static route must remain invalid/inactive. When an ARP entry for the next-hop is populated based on a gratuitous ARP received or

periodic traffic destined for it and the usual ARP who-has procedure, the static route becomes valid/active and is installed.

#### 2.5.1.1 Invalidate next-hop based on neighbor cache state

This feature invalidates a static route based on the reachability of the next-hop in the neighbor cache when the **validate-next-hop** command is enabled for an IPv6 static route.

Use the commands in the following contexts to enable validate-next-hop for an IPv4 static route:

- **MD-CLI**

```
configure router static-routes route next-hop
configure service vprn static-routes route next-hop
```

- **classic CLI**

```
configure router static-route-entry next-hop
configure service vprn static-route-entry next-hop
```

In this case, when the Neighbor Cache entry for next-hop is INVALID or not populated, the static route must remain invalid/inactive. When an NC entry for next-hop is populated based on a neighbor advertisement received, or periodic traffic destined for it and the usual NS/NA procedure, the static route becomes valid/active and is installed.

## 2.6 IP interface strip-label behavior

The strip-label feature causes arriving MPLS encapsulated traffic to be stripped of all MPLS labels (up to five) before processing the packet through Policy Based Routing (PBR) filters. Use the following command to configure strip-label.

```
configure router interface strip-label
```

If the packets do not have an IP header immediately following the MPLS label stack after the strip, they are discarded. Only MPLS encapsulated IP, IGP shortcuts, and VPRN over MPLS packets are processed. However, IPv4 and IPv6 packets that arrive without any labels are supported on an interface with strip label enabled.

The **strip-label** command operates in promiscuous mode. The router does not filter on the destination MAC address of the Ethernet frames. In some network designs, multiple ports may be tapped and combined into an interface toward the router. Promiscuous mode allows all of these flows to be processed without requiring the destination MAC address to be updated to match the router address.

To associate an interface that is configured with the **strip-label** command with a port, the port must be configured as single-fiber.

Packets subject to the strip-label action and mirrored (using mirrors or Lawful Intercept) contain the original MPLS labels (and other Layer 2 encapsulation) in the mirrored copy of the packet, as they appeared on the wire when the **mirror-dest** type is the default type “ether”. If the **mirror-dest** type is “ip-only”, the mirrored copy of the packet does not contain the original Layer 2 encapsulation or the stripped MPLS labels.

This command is supported on:

- optical ports for the 7750 SR and 7450 ESS
- null/dot1q encaps
- network ports
- IPv4
- IPv6

## 2.7 LDP shortcut for IGP route resolution

This feature enables you to forward user IP packets and specified control IP packets using LDP shortcuts over all network interfaces in the system that participate in the IS-IS and OSPF routing protocols. The default is to disable the LDP shortcut across all interfaces in the system.

Use the following commands to use LDP shortcuts for IGP route resolution:

- **MD-CLI**

```
configure router ldp ldp-shortcut ipv4
configure router ldp ldp-shortcut ipv6
```

- **classic CLI**

```
configure router ldp-shortcut [ipv4][ipv6]
```

### 2.7.1 IGP route resolution

When LDP shortcut is enabled, LDP populates the RTM with next-hop entries corresponding to all prefixes for which it activated an LDP FEC. For an activated prefix, two route entries are populated in RTM. One corresponds to the LDP shortcut next-hop and has an owner of LDP. The other one is the regular IP next-hop. The LDP shortcut next-hop always has preference over the regular IP next-hop for forwarding user packets and specified control packets over a specific outgoing interface to the route next-hop.

The prior activation of the FEC by LDP is done by performing an exact match with an IGP route prefix in RTM. It can also be done by performing a longest prefix match with an IGP route in RTM if the aggregate-prefix-match command option is enabled globally in LDP.

The LDP next-hop entry is not exported to the LDP control plane or to any other control plane protocols except OSPF, IS-IS, and an OAM control plane specified in Handling of control packets.

This feature is not restricted to /32 IPv4 prefixes or /128 IPv6 FEC prefixes. However, only /32 IPv4 and /128 IPv6 FEC prefixes are populated in the tunnel table for use as a tunnel by services.

All user and specified control packets for which the longest prefix match in RTM yields the FEC prefix are forwarded over the LDP LSP. The following is an example of the resolution process.

Assume that the egress LER advertised a FEC for some /24 prefix using the fec-originate command. At the ingress LER, LDP resolves the FEC by checking in RTM that an exact match exists for this prefix. After the LDP activates the FEC, it programs the NHLFE in the egress datapath and the LDP tunnel information in the ingress datapath tunnel table.

Next, LDP provides the shortcut route to RTM, which associates it with the same /24 prefix. There are two entries for this /24 prefix: the LDP shortcut next-hop and the regular IP next-hop. The latter was used by

LDP to validate and activate the FEC. RTM then resolves all user prefixes that succeed a longest prefix match against the /24 route entry to use the LDP LSP.

Now assume that the aggregate-prefix-match was enabled and that LDP found a /16 prefix in RTM to activate the FEC for the /24 FEC prefix. In this case, RTM adds a new, more-specific route entry of /24 and has the next-hop as the LDP LSP. However, RTM does not have a specific /24 IP route entry. RTM then resolves all user prefixes that succeed a longest prefix match against the /24 route entry to use the LDP LSP. All other prefixes that succeed a longest prefix match against the /16 route entry uses the IP next-hop. LDP shortcut also works when using RIP for routing.

### 2.7.2 LDP-IGP synchronization

See the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide* for information about LDP-IGP synchronization.

### 2.7.3 LDP shortcut forwarding plane

After the LDP activates an FEC for a prefix and programs RTM, it also programs the ingress tunnel table in IOM or online cards with the LDP tunnel information.

When an IPv4 packet is received on an ingress network interface, a subscriber IES interface, or a regular IES interface, the lookup of the packet by the ingress IOM or line card results in the packet being sent labeled with the label stack corresponding to the NHLFE of the LDP LSP when the preferred RTM entry corresponds to an LDP shortcut.

If the preferred RTM entry corresponds to an IP next-hop, the IPv4 packet is forwarded unlabeled.

The switching from the LDP shortcut next-hop to the regular IP next-hop when the LDP FEC becomes unavailable depends on whether the next-hop is still available. If it is (for example, the LDP FEC was withdrawn because of LDP control plane issues) the switchover should be faster. If the next-hop determination requires IGP to re-converge, this takes longer. However, no target is set.

The switching from a regular IP next-hop to an LDP shortcut next-hop usually occurs only when both are available. However, the programming of the NHLFE by LDP and the programming of the LDP tunnel information in the ingress IOM or line cards tunnel table are asynchronous. If the tunnel table is configured first, it is possible that traffic is black-holed for some time.

### 2.7.4 ECMP considerations

When ECMP is enabled and multiple equal-cost next-hops exist for the IGP route, the ingress IOM or line card sprays the packets for this route based on the hashing routine currently supported for IPv4 packets.

When the preferred RTM entry corresponds to an LDP shortcut route, spraying is performed across the multiple next-hops for the LDP FEC. The FEC next-hops can either be direct link LDP neighbors or T-LDP neighbors reachable over RSVP LSPs, in the case of LDP-over-RSVP, but not both. This is as per ECMP for LDP.

When the preferred RTM entry corresponds to a regular IP route, spraying is performed across regular IP next-hops for the prefix.

Spraying across regular IP next-hops and LDP-shortcut next-hops concurrently is not supported.

### 2.7.5 Handling of control packets

All control plane packets do not see the LDP shortcut route entry in RTM with the exception of the following control packets, which are forwarded over an LDP shortcut when enabled:

- A locally generated or in transit ICMP ping and trace route of an IGP route. The transit message appears as a user packet to the ingress LER node.
- A locally generated response to a received ICMP ping or trace route message.

All other control plane packets that require an RTM lookup and knowledge of which destination is reachable over the LDP shortcut continues to be forwarded over the IP next-hop route in RTM.

### 2.7.6 Handling of multicast packets

Multicast packets cannot be forwarded or received from an LDP LSP. This is because there is no support for the configuration of such an LSP as a tunnel interface in PIM. Only an RSVP P2MP LSP is currently allowed.

If a multicast packet is received over the physical interface, the uRPF check does not resolve to the LDP shortcut because the LDP shortcut route in RTM is not made available to multicast application.

### 2.7.7 Interaction with BGP route resolution to an LDP FEC

There is no interaction between an LDP shortcut for BGP next-hop resolution and the LDP shortcut for IGP route resolution. BGP continues to resolve a BGP next-hop to an LDP shortcut if the user enabled the following command option in BGP. The following example shows the configuration to enable the resolution of a BGP next-hop to an LDP shortcut.

**Example: MD-CLI**

```
[ex:/configure router "Base" bgp next-hop-resolution shortcut-tunnel]
A:admin@node-2# info
    family ipv4 {
        resolution-filter {
            ldp true
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>router>bgp>next-hop-res>shortcut-tunn# info
----------------------------------------------
                family ipv4
                    resolution-filter
                        ldp
                    exit
                exit
----------------------------------------------
```

### 2.7.8 Interaction with static route resolution to an LDP FEC

A static route continues to be resolved by searching an LDP LSP whose FEC prefix matches the specified indirect next-hop for the route. In contrast, the LDP shortcut for IGP route resolution uses the LDP LSP as a route. The most specific route for a prefix is selected and, if both a static and IGP routes exist, the RTM route type preference is used to select one.

### 2.7.9 LDP control plane

For the LDP shortcut to be usable, SR OS must originate a <FEC, label> binding for each IGP route it learns of even if it did not receive a binding from the next-hop for that route. The router must assume that it is an egress LER for the FEC until the route disappears from the routing table or the next-hop advertises a binding for the FEC prefix. In the latter case, SR OS becomes a transit LSR for the FEC.

SR OS originates a <FEC, label> binding for its system interface address only by default. The only way to originate a binding for local interfaces and routes that are not local to the system is by using the fec-originate capability.

Use the **fec-originate** command to generate bindings for all non-local routes for which this node acts as an egress LER for the corresponding LDP FEC. Specifically, this feature must support the FEC origination of IGP learned routes and subscriber/host routes statically configured or dynamically learned over subscriber IES interfaces.

An LDP LSP used as a shortcut by IPv4 packets may also be tunneled using the LDP-over-RSVP feature.

## 2.8 Weighted load-balancing over interface next-hops

When the **weighted-ecmp** command is configured in the base router context or a VPRN, any IPv4 or IPv6 static, IS-IS, or OSPF route associated with the routing instance can be programmed into the datapath to use weighted load-balancing across the interface next-hops of the route.

Use the following commands to configure weighted ECMP in the base router context or in a VPRN.

```
configure router weighted-ecmp
configure service vprn weighted-ecmp
```

In order for weighted ECMP to be supported across the interface next-hops of an IS-IS or OSPF route the following conditions must be met.

- All of the calculated ECMP next-hops must be interface next-hops.
- All of the calculated ECMP next-hop interfaces must have a non-zero load-balancing-weight value configured in the following context. Use the commands in the following context to configure a non-zero load-balancing-weight value.

  ```
  configure router isis interface
  ```

  By default, IS-IS or OSPF interfaces have a zero weight (no load-balancing-weight); non-zero values must be configured explicitly. Values cannot be auto-derived.

In order for weighted ECMP to be supported across the interface next-hops of a static route the following conditions must be met.

- All of the configured ECMP next-hops must be direct next-hops (resolved to an interface). The ECMP next-hops are the next-hops with the lowest preference that also have the lowest metric.
- All of the configured ECMP next-hop interfaces must have a non-zero load-balancing-weight value configured in the following context. Use the commands in the following context to configure a non-zero load-balancing-weight value:
  - **MD-CLI**

    ```
    configure router static-routes route next-hop
    ```

  - **classic CLI**

    ```
    configure router static-route-entry next-hop
    ```

  By default, static route next-hops have a zero weight (no load-balancing-weight); non-zero values must be configured explicitly. Values cannot be auto-derived. The ECMP next-hops are the next-hops with the lowest preference that also have the lowest metric.

The **load-balancing-weight** commands in the IS-IS or OSPF and static route configuration trees accept a value between 0 and 4294967295.

If an IPv4 or IPv6 BGP route has a BGP next-hop resolved by a static, IS-IS, or OSPF ECMP route and **ibgp-multipath** is configured under BGP, traffic forwarded to the BGP next-hop is sprayed according to the load-balancing-weights of the interface next-hops.

## 2.9 IP FRR for static route entry

IP Fast ReRoute (FRR) is supported when the **backup-next-hop** command is configured for a static route entry. IP FRR support uses 1+1 protection by using a single backup next-hop address when the single primary next-hop fails. Only 1+1 protection is supported during backup without ECMP capability. Next-hop forwarding information for the backup next-hop address from the IP Routing Table Manager (RTM) is used to install a pre-resolved IP or tunneled fast reroute backup path to the backup next-hop. The configured backup next-hop IP address can be directly or indirectly connected through an IGP, a BGP, or a tunnel. The backup next-hop must be of the same IP address family as the primary next-hop (for example, an IPv4 primary next-hop can be protected using an IPv4 backup next-hop).

**Note:** FRR for static route entries is only supported for IP traffic on FP-based platforms.

IP FRR for static route is supported in the base router and service VPRN contexts.

If the primary next-hop of the static route entry fails and the IP FRR backup next-hop is activated, then the backup tag is applied to the static route and the configured preference and metric for the primary hop is inherited. If the primary next-hop is activated again, then make-before-break functionality is used to avoid any packet loss.

The following example shows the IP FRR configuration.

### Example: MD-CLI

```
[ex:/configure router "Base"]
A:admin@node-2# info
    static-routes {
        route 10.10.0.0/16 route-type unicast {
```

```
            tag 20
            backup-tag 100
            next-hop "101.1.1.1" {
                preference 100
                backup-next-hop {
                    address 50.1.1.2
                }
            }
        }
    }
```

### Example: classic CLI

```
A:node-2>config>router# info
#------------------------------------------------
"Static Route Configuration"
#------------------------------------------------
        static-route-entry 10.10.0.0/16
            tag 20
            backup-tag 100
            next-hop 101.1.1.1
                preference 100
                backup-next-hop
                    address 50.1.1.2
                exit
            exit
        exit
```

The logic behavior applied to the associated tag of the static route entry is summarized in the following table.

*Table 3: Static route tag for IP FRR configuration*

| Primary NH | Backup NH | StaticRoute State | StaticRoute Tag |
|---|---|---|---|
| UP | UP | UP | 20[1] |
| UP | DOWN | UP | 20[1] |
| DOWN | UP | UP | 100[1] |
| DOWN | DOWN | DOWN | — |

IGP export policies can use the tag and the backup-tag as match criteria when exporting a static route entry using route policies. The export policies may introduce unique export properties for each tag (for example, resulting in different IGP metrics) and may make an exported route more or less desirable when the primary next-hop fails and the backup next-hop is activated.

The following limitations apply in the IP FRR for static route entries.

- Only the primary next-hop has IP FRR support. The backup next-hop has no IP FRR support if it suddenly becomes unreachable.
- If multiple next-hops are configured with a backup for a static route entry, then IP FRR is activated if there is only one remaining primary next-hop active. If multiple primary next-hops can be activated, then the static route entry uses ECMP and the backup next-hop IP FRR functionality is not used.

[1] The tag value is based on the preceding IP FRR example configuration provided.

- If the primary next-hop fails and the backup next-hop is used as the primary hop, then the backup next-hop uses the configured backup tag (or 0, if not configured) and inherits the configured preference and metric of the primary next-hop (or the default values, if not configured).
- The backup inherits the preference and the metric of the primary next-hop, however, it does not support any of the features configured on the primary next-hop (for example, BFD, CPE check, LDP sync, and so on) even when the backup becomes the active next-hop.
- If the primary next-hop of a static route entry, configured with a backup next-hop, is held down because hold-down is configured on static routes, the backup next-hop is also held down and is not used for traffic, even in cases where the backup-next-hop can be activated.
- The following tunnel types are supported:
  - OSPF or ISIS shortcuts using RSVP-TE and SR-TE
  - BGP VPN-v4/v6 or BGP shortcut routes over LDP, RSVP, SR-ISIS, SR-OSPF, LDPoRSVP, SR-TE, GREv4, SR policy, MPLS forward policy, and RIB API
  - backup-next-hop recursion through indirect next-hop static-route-entry with resolution filter for LDP, RSVP, LDPoRSVP, SR-TE, SR-ISIS, SR-OSPF, SR policy, MPLS forward policy, or RIB API
- LDP-FRR using a static-route is not mutually supported in combination with static-route backup-next-hop for the same static route.
- Any other backup-next-hop types are considered as non-supported. For example:
  - Locally aggregated BGP routes
  - BGP routes when the BGP next-hop is recursively resolved through another BGP route
  - 6over4 tunnel
  - GREv6 tunnel
  - OSPF or IS-IS shortcuts using LDP, SR-ISIS, SR-OSPF, and LDPoRSVP (generic IGP shortcut limitation not only for backup-next-hop)
  - OSPF or IS-IS shortcuts over SR policy, MPLS forward policy, and RIB API (generic IGP shortcut limitation not only for backup-next-hop)
  - BGP-LU over LDP, RSVP, LDPoRSVP, SR-TE, SR-OSPF, SR-ISIS, SR policy, MPLS forward policy and RIB API
  - 4PE
  - 6PE

## 2.10 IP-over-GRE and MPLS-over-GRE termination on a user-configured subnet

This feature enables the termination of MPLS-over-GRE and IP-over-GRE packets on destination IP addresses from a user-configured subnet. SR OS supports processing received GRE encapsulated packets concurrently when the destination address in the outer IPv4 header matches the system interface address (exact match) and when it matches an address on the user-configured GRE termination subnet (longest prefix match).

RFC 2890 specifies the following format for the GRE header:

```
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|C| |K|S| Reserved0       | Ver |         Protocol Type         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Checksum (optional)      |       Reserved1 (Optional)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Key (optional)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Sequence Number (Optional)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

All the fields of the GRE encapsulation in RFC 2890 are optional except for the base header (first 4 bytes). The C, K, and S flags are used to indicate if the header includes the optional fields of Checksum (plus Reserved field), Key, and Sequence Number. SR OS can process packets received with the base 4-byte header or with the 8-byte header which includes the Key field. In other words, packets with the flags set to {C=0, K=0/1, S=0}. Any other GRE header setting results in the packet being dropped.

When originating a GRE encapsulated packet, SR OS supports the following header formats:

1. The 4-byte base header {C=0, K=0, S=0} in the IP-over-GRE feature using a Port Cross Connect (PXC) port (see GRE tunnel overview).
2. The 4-byte base header {C=0, K=0, S=0} in the IP-over-GRE feature using the Multiservice Integrated Service Adapter.

   See *Section 4.1, IP Tunnel Overview*, of the *7450 ESS, 7750 SR, and VSR Multiservice ISA and ESA Guide*.
3. The 4-byte base header {C=0, K=0, S=0} in the MPLS-over-GRE tunnel and SDP.

   See *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 2 Services and EVPN Guide*, the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Guide: IES and VPRN*, and the *7450 ESS, 7750 SR, 7950 XRS, and VSR Services Overview Guide*.
4. The 8-byte header which includes the Key field {C=0, K=1, S=0} in the filter-based GRE tunneling feature (see Configuring filter-based GRE tunneling ).

### 2.10.1 Feature configuration

The user defines a subnet for the termination of GRE packets by applying the **gre-termination** command to a numbered network IP interface, including a loopback interface. Use the following command to configure GRE termination.

```
configure router interface gre-termination
```

The following rules apply to termination of IP-over-GRE and MPLS-over-GRE on a user-defined subnet.

- The termination of MPLS-over-GRE on the system interface address can be performed concurrently and extends to terminating IP-over-GRE packets as well.
- A single GRE termination subnet is permitted per router. If the user attempts to configure another subnet on another interface, the command is rejected.
- The GRE termination subnet length can be of maximum size of /16.
- The subnet of the primary IPv4 address of the numbered loopback interface or the numbered network IP interface is used as the GRE termination subnet.

- When the GRE termination subnet is enabled on a numbered network IP interface, the packet can be received from the interface itself and any other network IP interface as long as the target IPv4 termination subnet is reachable.
- The feature can terminate packets with the base 4-byte header {C=0, K=0, S=0} or with the 8-byte header which includes the Key field {C=0, K=1, S=0}. Any other GRE header setting results in the packet being dropped.
- For routers in the network to forward MPLS-over-GRE or IP-over-GRE packets to this router, the prefix of the GRE subnet must be advertised in IGP or BGP. This is performed by adding the interface to IGP or BGP. Alternatively, a static route is added to the other routers.
- The GRE termination subnet is not supported with the following interface types. If these interface types are configured, the configuration of the **gre-termination** command option is rejected:
  - unnumbered network IP interface
  - IES interface
  - VPRN interface
  - CSC VPRN interface
- The configuration of the **gre-termination** command option is also rejected when applied to the system interface, as the system interface supports the termination of MPLS-over-GRE packet on its /32 subnet with no explicit configuration.
- This feature introduces full support of LER and LSR roles for the packet after the GRE encapsulation is removed, regardless if the GRE termination was on the system interface address or the GRE termination subnet.
- In an LSR role, this feature sprays the decapsulated packets over LAG and ECMP links by attempting a hash on the SA/DA and Layer 4 ports of the inner IP header if the payload below the label stack is IPv4 or IPv6. Otherwise, a hash is performed on the SA/DA of the outer IPv4 header of the GRE encapsulation.

### 2.10.2 MPLS-over-GRE and IP-over-GRE termination function

When a GRE packet is received over any network IP interface, the router checks if destination address matches the system interface address (exact match) or the GRE termination subnet (Longest Prefix Match). The router then processes the packet according to the following criteria:

- If a match exists and the GRE Protocol Type field indicates an MPLS payload, continue processing the MPLS label stack as normal. This includes:
  - Pop one or more labels and forward to CPM if a MPLS exception exists (TTL expiry, RA label, 127/8 destination address in underlying IP packet).
  - Pop one or more labels and look up the packet in the FIB or in a local service context. The router operates as an egress LER.
  - Pop one or more labels and swap a label out to the outgoing interface with NHLFE encapsulation pushed on the packet. The router operates as an LSR.
  - When the incoming label is swapped to an implicit-null label, the user is able to remark the DSCP field of the exposed IPv4 or IPv6 packet on egress of the datapath.
- If a match exists and the GRE Protocol Type field indicates an IPv4 or an IPv6 payload, continue processing in the pipeline as an IP packet and forward out based on FIB lookup.

- If a match exists and the GRE Protocol Type field indicates a Bridged Ethernet payload, drop the packet. To enable the feature to terminate the Bridged Ethernet payload, ensure that the termination subnet for that feature does not overlap with the GRE termination subnet of MPLS-over-GRE and IP-over-GRE termination.
- If a match exists and the GRE protocol Type field is set to any other payload value, drop the packet.
- If a match exists and the packet is not dropped, the application of ACL filter on the incoming interface matches against the inner (payload) header of the received GRE-encapsulated packet.
- If a match does not exist, continue processing in the pipeline as an IPv4 packet. In this case, the application of ACL filter on the incoming interface matches against the outer IPv4 header of the received GRE-encapsulated packet.

This feature supports GRE/IPv4 encapsulation when the payload is MPLS, IPv4, or IPv6.

All MPLS egress LER and LSR features associated with the processed label are supported.

### 2.10.3 Outgoing packet Ethertype setting and TTL handling in MPLS-over-GRE termination

The router sets the Ethertype field value of the outgoing packet according to the following criteria.

- If the swapped label is not the Bottom-of-Stack label, Ethertype is set to MPLS value.
- If the swapped label is the Bottom-of-Stack label and the outgoing label is not implicit-null, Ethertype is set to MPLS value.
- If the swapped label is the Bottom-of-Stack label and the outgoing label is implicit-null, Ethertype is set to IPv4 or IPv6 value when the first nibble of the exposed IP packet is 4 or 6 respectively. If the first nibble value is neither 4 nor 6, the packet is dropped.

The router sets the TTL of the outgoing packet as per the behavior of a PHP LSR:

- The TTL of a forwarded IP packet is set to MIN (MPLS_TTL-1, IP_TTL), where MPLS_TTL refers to the TTL in the outermost label in the popped stack and IP_TTL refers to the TTL in the exposed IP header.
- The TTL of a forwarded MPLS packet is set to MIN(MPLS_TTL-1, INNER_MPLS_TTL), where MPLS_TTL refers to the TTL in the outermost label in the popped stack and INNER_MPLS_TTL refers to the TTL in the exposed label.

### 2.10.4 Ethertype setting and TTL handling in IP-over-GRE termination

The router sets the Ethertype field value of the outgoing packet to IPv4 or IPv6 value when the GRE protocol field value in the incoming packet is IPv4 or IPv6 respectively.

The router checks and decrements the TTL field of the inner IPv4 or IPv6 header and ignores the TTL of the outer IPv4 header.

### 2.10.5 LER and LSR hashing support

When the router removes the GRE encapsulation, pops one or more labels including the Bottom-of-Stack (BoS) label, it acts as a LER. The exposed packet are forwarded in the global routing table or in a service context. The LAG/ECMP hashing of the packet when forwarded follow the procedures of that specific

forwarding context. see "Traffic Load Balancing Options" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Interface Configuration Guide*.

When the router removes the GRE encapsulation, pops one or more labels and then swaps a label, it acts as an LSR. The LSR hashing for packets of a MPLS-over-GRE SDP or tunnel terminating on the GRE subnet follows a new procedure which is enabled automatically and overrides the LSR hashing command option enabled on the incoming network IP interface. Use the following command to configure the LSR hashing command option.

```
configure router interface load-balancing lsr-load-balancing
```

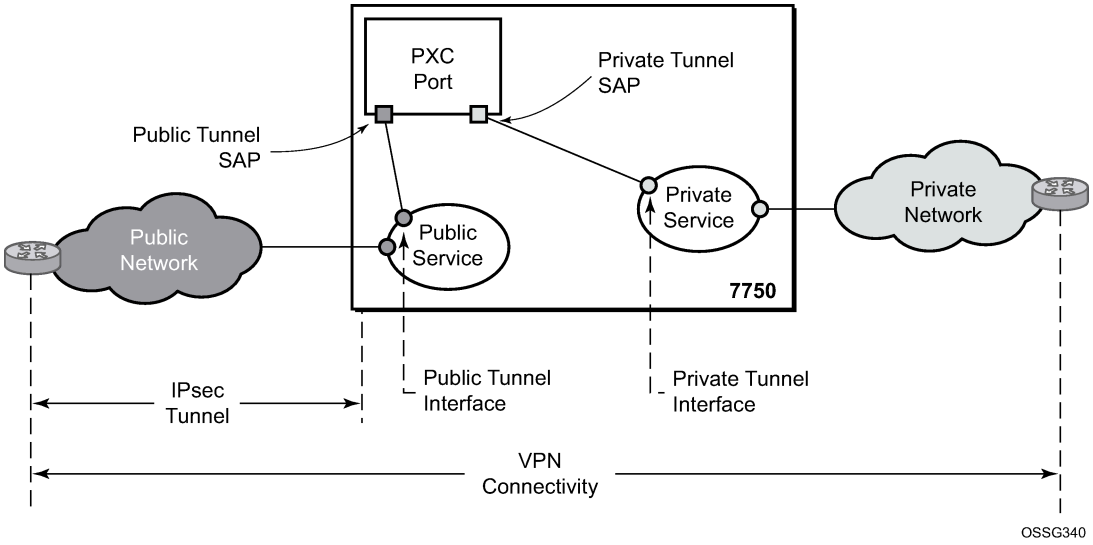
For more details, see *LSR Hashing of MPLS-over-GRE Encapsulated Packet* in section *Changing Default Per Flow Hashing Inputs* of the *7450 ESS, 7750 SR, 7950 XRS, and VSR Interface Configuration Guide*.

## 2.11 GRE tunnel overview

This section describes the GRE tunneling feature supported through the use of a Port Cross Connect (PXC) port. In this application, the PXC port functions as a resource module for the system, providing the necessary resources for the GRE encapsulation function. The GRE encapsulation function described here is similar to the GRE tunnel functionality supported through the use of the MS-ISA. In this use case, the MS-ISA is not required.

Figure 11: GRE deployment using a PXC port example shows an example of a GRE deployment supported inside a 7750 SR router using the PXC element.

*Figure 11: GRE deployment using a PXC port example*




In Figure 11: GRE deployment using a PXC port example, the public network is typically an unsecured network, such as public Internet, over which packets belonging to the private network in the diagram cannot be transmitted natively. Inside the 7750 SR, a public service instance (IES or VPRN) connects to the public network, and a private service instance (typically a VPRN) connects to the private network.

For GRE tunnels using PXC ports, the public and private services must be two different services, and the PXC is the connection between the two services. Traffic from the public network may require

authentication and encryption inside an IPsec tunnel to reach the private network. In this way, the authenticity, confidentiality, and integrity of private network access can be enforced. If authentication and confidentiality are not required, then access to the private network may be provided through GRE or IP-IP tunnels.

Traffic flows through PXC-based tunnels in the following ways:

- In the upstream direction (public to private), the encapsulated traffic is forwarded to a public tunnel interface if the destination address matches the local or gateway address of a GRE tunnel. As the traffic passes through the PXC port, the tunnel header is removed, the payload IP packet is delivered to the private service, and from there, the traffic is forwarded again based on the destination address of the payload IP packet.
- In the downstream direction (private to public), unencapsulated traffic belonging to the private service is forwarded into the tunnel by matching a route with the GRE tunnel as next-hop. The route can be configured statically, learned by running OSPF on the private tunnel interface or by running BGP over the tunnel. After clear traffic is forwarded to the PXC port, it is encapsulated in the GRE header and passed to the public service, and from there, the traffic is forwarded again based on the destination address of the GRE header.

### 2.11.1 Example GRE tunnel configurations

The following example shows the configuration of a public interface.

**Example: Public interface configuration (MD-CLI)**

```
[ex:/configure service ies "100"]
A:admin@node-2# info
    admin-state enable
    customer "1"
    interface "int-gre-tunnel-public" {
        sap pxc-1.b:100 {
            description "Public Tunnel PXC SAP"
        }
        ipv4 {
            primary {
                address 192.110.1.1
                prefix-length 30
            }
        }
    }
```

**Example: Public interface configuration (classic CLI)**

```
A:node-2>config>service# info
----------------------------------------------
        ies 100 name "100" customer 1 create
            no shutdown
            interface "int-gre-tunnel-public" create
                address 192.110.1.1/30
                sap pxc-1.b:100 create //Public interface
                    description "Public Tunnel PXC SAP"
                exit
            exit
            no shutdown
----------------------------------------------
```

The following example shows the configuration of a private interface.

### Example: Private interface configuration (MD-CLI)

```
[ex:/configure service]
A:admin@node-2# info
    customer "200" {
    }
    vprn "200" {
        customer "200"
        bgp-ipvpn {
            mpls {
                admin-state enable
                route-distinguisher "64496:1"
                vrf-target {
                    community "target:64496:1"
                }
            }
        }
        interface "int-gre-tunnel-private" {
            tunnel true
            ip-mtu 1476
            ipv4 {
                addresses {
                    address 10.1.1.1 {
                        prefix-length 30
                    }
                }
            }
            sap pxc.1.a:200 {
                ip-tunnel "gre-tunnel-1" {
                    admin-state enable
                    delivery-service "100"
                    remote-ip-address 192.120.1.1
                    backup-remote-ip-address 192.120.1.2
                    local-ip-address 192.110.1.2
                    gre-header {
                        admin-state enable
                        key {
                            admin-state enable
                            send 123
                            receive 123
                        }
                    }
                }
            }
        }
        static-routes {
            route 172.16.1.1/24 route-type unicast {
                next-hop "10.1.1.2" {
                }
            }
        }
    }
```

### Example: Private interface configuration (classic CLI)

```
A:node-2>config>service# info
----------------------------------------------
        customer 200 name "200" create
        exit
        vprn 200 name "200" customer 200 create
            no shutdown
            interface "int-gre-tunnel-private" tunnel create
                address 10.1.1.1/30
```

```
                ip-mtu 1476
                sap pxc.1.a:200 create
                    ip-tunnel "gre-tunnel-1" create
                        gre-header send-key 123 receive-key 123
                        source 192.110.1.2
                        remote-ip 192.120.1.1
                        backup-remote-ip 192.120.1.2
                        delivery-service name "100"
                        no shutdown
                    exit
                exit
            exit
            static-route-entry 172.16.1.1/24
                next-hop 10.1.1.2
                    no shutdown
                exit
            exit
            bgp-ipvpn
                mpls
                    route-distinguisher 64496:1
                    vrf-target target:64496:1
                    no shutdown
                exit
            exit
        exit
-------------------------------------------
```

## 2.12 Router interface encryption with NGE

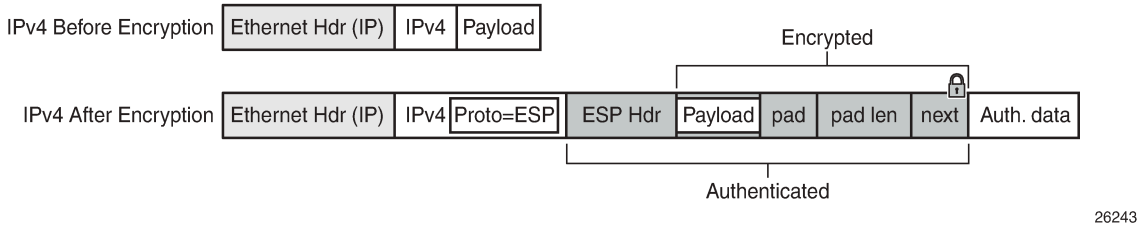
NGE nodes support Layer 3 encryption on router interfaces for IPv4 traffic. NGE is not supported on dual-stack IPv4/IPv6 or IPv6-only interfaces. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR Services Overview Guide* for more information about platforms that support NGE.

NGE is enabled on a router interface by configuring the **group-encryption** command on the router interface. The interface is considered part of the NGE domain, and any received packets that are NGE-encrypted are decrypted if the key group is configured on the node. To encrypt packets egressing the interface, the outbound key group must be configured on the interface. All IP packets, such as self-generated traffic or packets forwarded from router interfaces that are not inside the NGE domain, are encrypted when egressing the interface. There are some exceptions to this general behavior, as described in the sections that follow; for example, GRE-MPLS and MPLSoUDP packets are not encrypted when router interface encryption is enabled.

The outbound and inbound key groups configured on the router interface determine which keys are used to encrypt and decrypt traffic. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR Services Overview Guide* for more information about configuring key groups.

To perform encryption, router interface encryption reuses the IPsec transport mode packet format as shown in Figure 12: Router Interface Encryption Packet Format (IPsec Transport Mode).

*Figure 12: Router Interface Encryption Packet Format (IPsec Transport Mode)*




The protocol field in the IP header of an NGE packet is always set to "ESP". Within an NGE domain, the SPI that is included in the ESP header is always an SPI for the key group configured on the router interface. Other fields in the IP header, such as the source and destination addresses, are not altered by NGE router interface encryption. Packets are routed through the NGE domain and decrypted when the packet leaves the NGE domain.

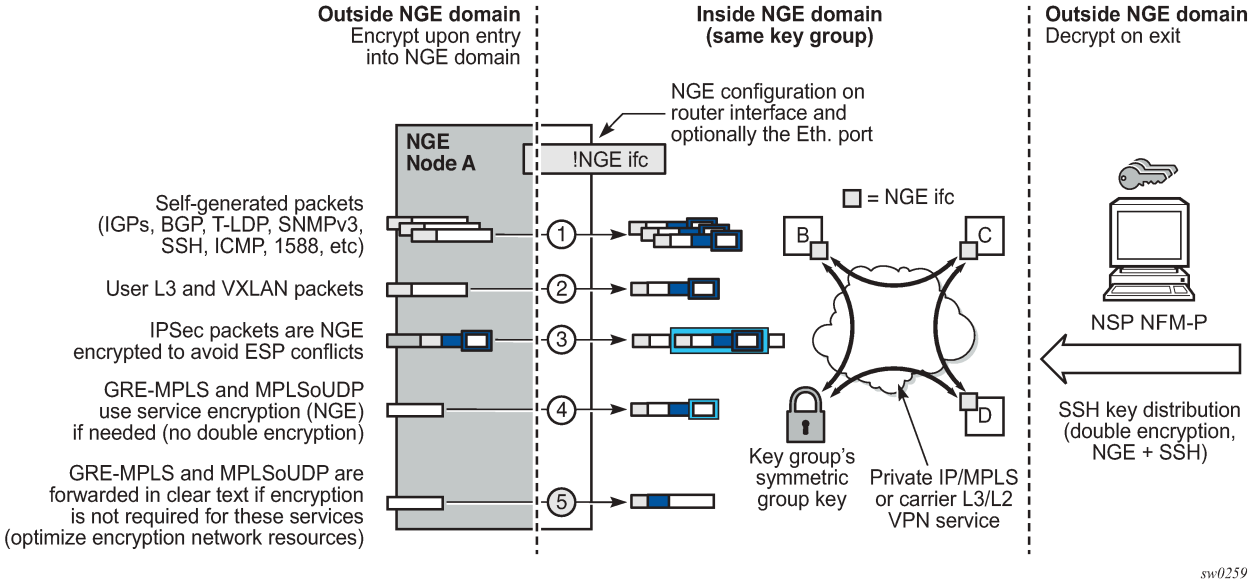
The group keys used on an NGE-enabled router interface provide encryption of broadcast and multicast packets within the GRT. For example, OSPF uses a broadcast address to establish adjacencies, which can be encrypted by NGE without the need to establish point-to-point encryption tunnels. Similarly, multicast packets are also encrypted without point-to-point encryption tunnels.

### 2.12.1 NGE domains

An NGE domain is a group of nodes and router interfaces forming a network that uses a single key group to create a security domain. NGE domains are created when router interface encryption is enabled on router interfaces that need to participate in the NGE domain. The NSP NFM-P assists users in managing the nodes and interfaces that participate in the NGE domain. See the *NSP NFM-P User Guide* for more information.

Figure 13: NGE Domain Transit shows various traffic types crossing an NGE domain.

*Figure 13: NGE Domain Transit*




In Figure 13: NGE Domain Transit, nodes A, B, C, and D have router interfaces configured with router interface encryption enabled. Traffic is encrypted when entering the NGE domain using the key group configured on the router interface and is decrypted when exiting the NGE domain. Traffic may traverse multiple hops before exiting the NGE domain, yet decryption only occurs on the final node when the traffic exits the NGE domain.

Various traffic types are supported and encrypted when entering the NGE domain, as illustrated by the following items on node A in Figure 13: NGE Domain Transit:

- Item 1: Self-generated packets

  These packets, which include all types of control plane and management packets such as OSPF, BGP, LDP, SNMPv3, SSH, ICMP, RSVP-TE, and 1588, are encrypted.

- Item 2: User Layer 3 and VXLAN packets

  Any Layer 3 user packets that are routed into the NGE domain from an interface outside the NGE domain are encrypted. Any VXLAN packets that are routed into the NGE domain from this NGE node are encrypted.

- Item 3: IPsec packets

  IPsec packets are NGE-encrypted when entering the NGE domain to ensure that the IPsec packets' security association information does not conflict with the NGE domain.

GRE-MPLS- or MPLSoUDP-based service traffic consists of Layer 3 packets, and router interface NGE is not applied to these types of packets. Instead, service-level NGE is used for encryption to avoid double-encrypting these packets and impacting throughput and latencies. The two types of GRE-MPLS or MPLSoUDP packets that can enter the NGE domain are illustrated by items 4 and 5 in Figure 13: NGE Domain Transit.

- Item 4: GRE-MPLS and MPLSoUDP packets (SDP or VPRN) with service-level NGE enabled

  These encrypted packets use the key group that is configured on the service. The services key group may be different from the key group configured on the router interface where the GRE-MPLS or MPLSoUDP packet enters the NGE domain.

- Item 5: GRE-MPLS and MPLSoUDP packets (SDP or VPRN) with NGE disabled

  These packets are not encrypted and can traverse the NGE domain in clear text. If these packets require encryption, SDP or VPRN encryption must be enabled.

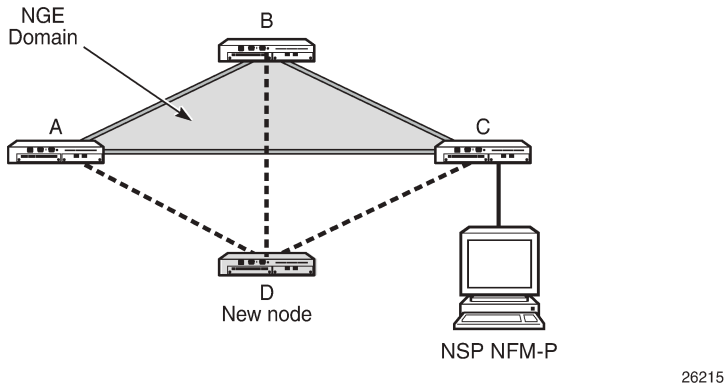
Creating an NGE domain from the NSP NFM-P requires the user to determine the type of NGE domain being managed. This indicates whether NGE gateway nodes are required to manage the NGE domain, and other operational considerations. The two types of NGE domains are:

- Private IP/MPLS network NGE domain
- Private over intermediary network NGE domain

### 2.12.1.1 Private IP/MPLS network NGE domain

One type of NGE domain is a private IP/MPLS network, as shown in Figure 14: Private IP/MPLS network NGE domain.

*Figure 14: Private IP/MPLS network NGE domain*



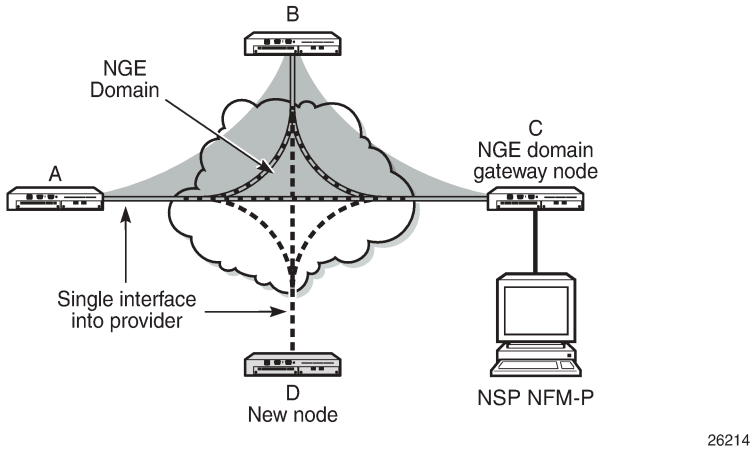

In a private IP/MPLS network NGE domain, all interfaces are owned by the user and there is no intermediary service provider needed to interconnect nodes. Each interface is a point-to-point private link between private nodes. When a new node is added to this type of NGE domain (node D in Figure 14: Private IP/MPLS network NGE domain), the links that connect node D to the existing nodes in the NGE domain (nodes A, B, and C) must be enabled with NGE router interface encryption. Links from the new node to the existing nodes are enabled one at a time. The NSP NFM-P provides tools that simplify adding nodes to the NGE domain and enabling NGE on their associated interfaces. In this type of NGE domain, each interface is a direct link between two nodes and is not used to communicate with multiple nodes over a broadcast medium offered by an intermediary network. Also, there are no NGE gateway nodes required between the NSP NFM-P and new nodes entering the NGE domain.

### 2.12.1.2 Private over intermediary network NGE domain

The other type of NGE domain is a private IP/MPLS network that traverses an intermediary network NGE domain; the intermediary network is used to interconnect nodes in the NGE domain using a multipoint-to-multipoint service. The intermediary network is typically a service provider network that provides a private IP VPN service or a private VPLS service used to interconnect a private network that does not mimic point-to-point links as described in the Private IP/MPLS network NGE domain section.

This type of NGE domain is shown in Figure 15: Private over intermediary network NGE domain.

*Figure 15: Private over intermediary network NGE domain*




Private over intermediary network NGE domains have nodes with links that connect to a service provider network where a single link can communicate with multiple nodes over a Layer 3 service such as a VPRN. In Figure 15: Private over intermediary network NGE domain, node A has NGE enabled on its interface with the service provider and uses that single interface to communicate with nodes B and C, and eventually with node D when node D has been added to the NGE domain. This type of NGE domain requires the recognition of NGE gateway nodes that allow the NSP NFM-P to reach new nodes that enter the domain. Node C is designated as a gateway node.

When node D is added to the NGE domain, it must first have the NGE domain key group downloaded to it from the NSP NFM-P. The NSP NFM-P creates an NGE exception ACL on the gateway node, C, to allow communication with node D using SNMPv3 and SSH through the NGE domain. After the key group is downloaded, the NSP NFM-P enables router interface encryption on node D's interface with the service provider and node D is now able to participate in the NGE domain. The NSP NFM-P automatically removes the IP exception ACL from node C when node D enters the NGE domain.

See Router interface NGE domain concepts for more information.

## 2.12.2 Router interface NGE domain concepts

An NGE domain is a group of nodes whose router interfaces in the base routing context (GRT) are enabled for router interface NGE. An interface without router interface NGE enabled is considered to be outside the NGE domain. NGE domains use only one key group when the domain is created; however, two key groups may be active at when if some links within the NGE domain are in transition from one key group to the other.

Figure 16: Inside and outside NGE domains illustrates the NGE domain concept. Table 4: Inside and outside NGE domains configuration scenarios describes the three configuration scenarios inside the NGE domain.

*Figure 16: Inside and outside NGE domains*

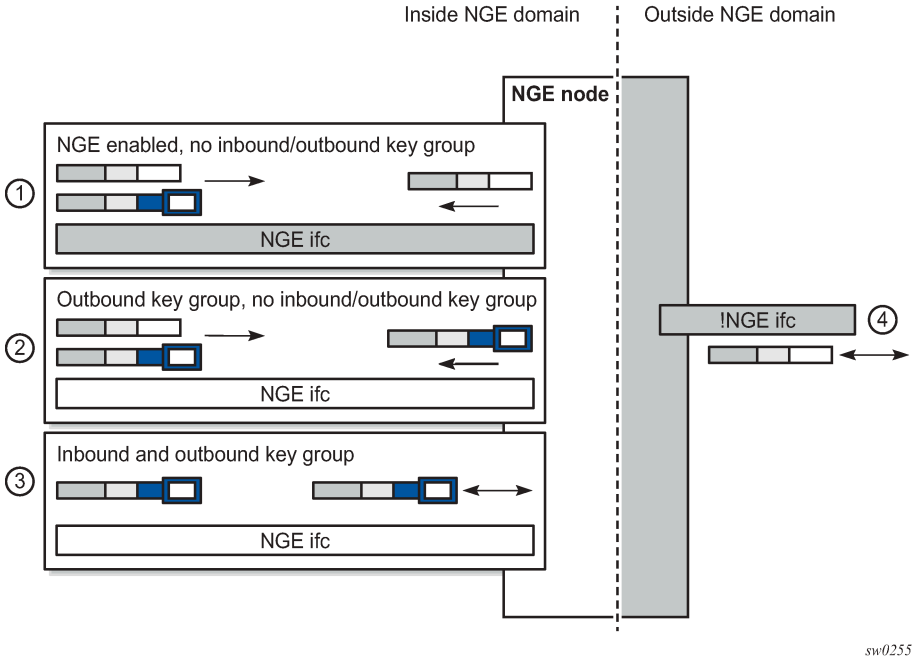



*Table 4: Inside and outside NGE domains configuration scenarios*

| Key | Description |
|---|---|
| 1 | NGE enabled, no inbound/outbound key group<br>Outbound packets are sent without encrypting; inbound packets can be NGE-encrypted or clear text |
| 2 | Outbound key group, no inbound key group<br>Outbound packets are encrypted using the interface key group if not already encrypted; inbound packets can be NGE-encrypted or clear text |
| 3 | Inbound and outbound key group<br>Outbound packets are encrypted using the interface key group if not already encrypted; inbound packets must be encrypted using the interface key group keys |
| 4 | Outside the NGE domain, the interface is not configured for NGE; any ESP packets are IPsec packets |

A router interface is considered to be inside the NGE domain when it has been configured with **group-encryption** on the interface. When **group-encryption** is configured on the interface, the router can receive unencrypted packets or NGE-encrypted packets from any configured key group on the router, but any other type of IPsec-formatted packet is not allowed. If an IPsec-formatted packet is received on an interface that has **group-encryption** enabled, it does not pass NGE authentication and is dropped. Therefore, IPsec

packets cannot exist within the NGE domain without first being converted to NGE packets. This conversion requirement delineates the boundary of the NGE domain and other IPsec services.

When NGE router interface encryption is enabled and only an outbound key group is configured, the interface can receive unencrypted packets or NGE-encrypted packets from any configured key group on the router. All outbound packets are encrypted using the outbound key group if the packet was not already encrypted further upstream in the network.

When NGE router interface encryption has been configured with both an inbound and outbound key group, only NGE packets encrypted with the key group security association can be sent and received over the interface.

When there is no NGE router interface encryption, the interface is considered outside the NGE domain where NGE is not applied.

See the "NGE Packet Overhead and MTU Considerations" section in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Services Overview Guide* for MTU information related to enabling NGE on a router interface.

### 2.12.3 GRE-MPLS and MPLSoUDP packets inside the NGE domain

NGE router interface encryption is never applied to GRE-MPLS or MPLSoUDP packets, for example:

- GRE with the GRE protocol ID set to MPLS Unicast (0x8847) or Multicast (0x8848)
- UDP packets with destination port = 6635)

GRE-MPLS and MPLSoUDP packets that enter the NGE domain or transit the NGE domain are forwarded as is.

Because these GRE-MPLS and MPLSoUDP packets provide transport for MPLS-based services, they already use the NGE services-based encryption techniques for MPLS, such as SDP or VPRN-based encryption. To avoid double encryption, the packets are left in cleartext when entering an NGE domain or crossing intermediate nodes in the NGE domain, and are forwarded as needed when exiting an NGE domain.

### 2.12.4 EVPN-VXLAN tunnels and services

NGE router interface encryption does not differentiate between EVPN-VXLAN tunnels and other L3 traffic, and therefore encrypts all EVPN-VXLAN traffic that egresses the node.

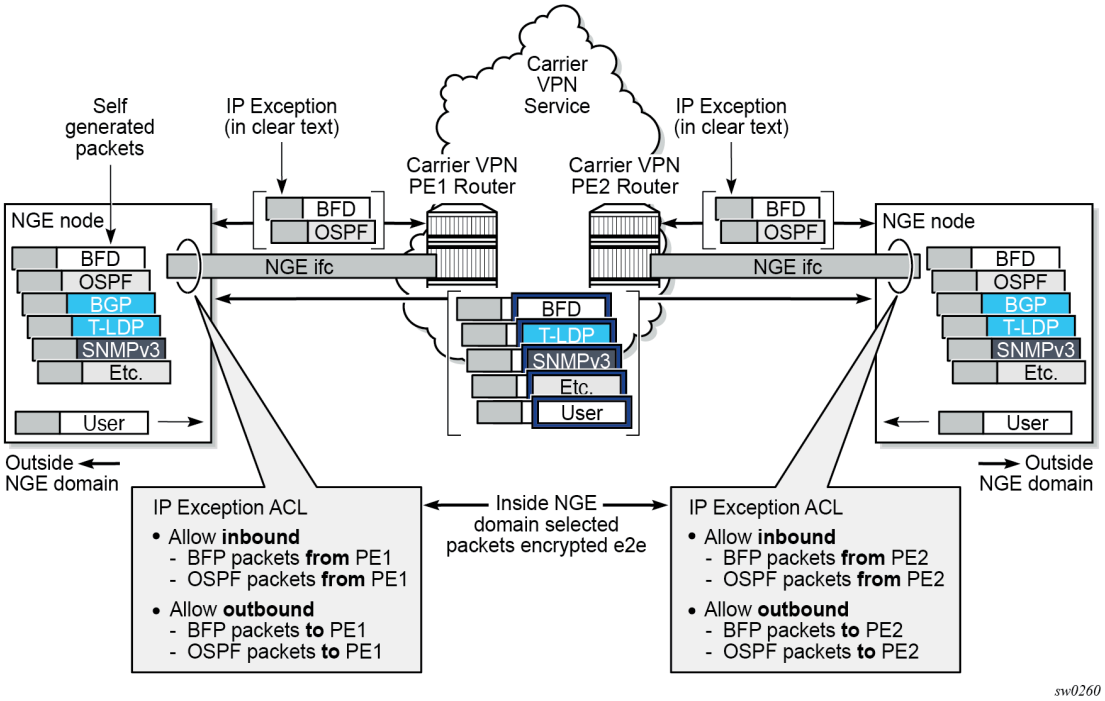
For received encrypted EVPN-VXLAN packets, if the VXLAN tunnel terminates on the node (that is, the destination IP is for a VTEP on this node), then the NGE packet is decrypted and the EVPN-VXLAN traffic is processed as if NGE encryption never took place.

### 2.12.5 Router encryption exceptions using ACLs

In some cases, Layer 3 packets may need to cross the NGE domain in clear text, such as when an NGE-enabled router needs to peer with a non-NGE-capable router to exchange routing information. This can be accomplished by using a router interface NGE exception filter applied on the router interface for the required direction, inbound or outbound.

Figure 17: Router interface NGE exception filter example shows the use of a router interface NGE exception filter.

*Figure 17: Router interface NGE exception filter example*




The inbound or outbound exception filter is used to allow specific packet flows through the NGE domain in clear text, where there is an explicit inbound and outbound key group configured on the interface. The behavior of the exception filter for each router interface configuration is as follows:

- NGE enabled, no inbound or outbound key group

  In this scenario, the router does not encrypt outbound traffic, and so the outbound exception filter is not applied. The router can still receive inbound NGE packets, so the exception filter is applied to inbound packets. If the filter detects a match, clear text packets can be received and forwarded by the router.

- Outbound key group, no inbound key group

  The outbound exception filter is applied to outbound traffic, and packets that match the filter are not encrypted on egress. The router can receive inbound NGE packets without an inbound key group set and applies the exception filter to inbound packets. If the filter detects a match, clear text packets can be received and forwarded by the router.

- Inbound and outbound key group

  The inbound and outbound exception filters are applied, and any packets that match are passed in clear text.

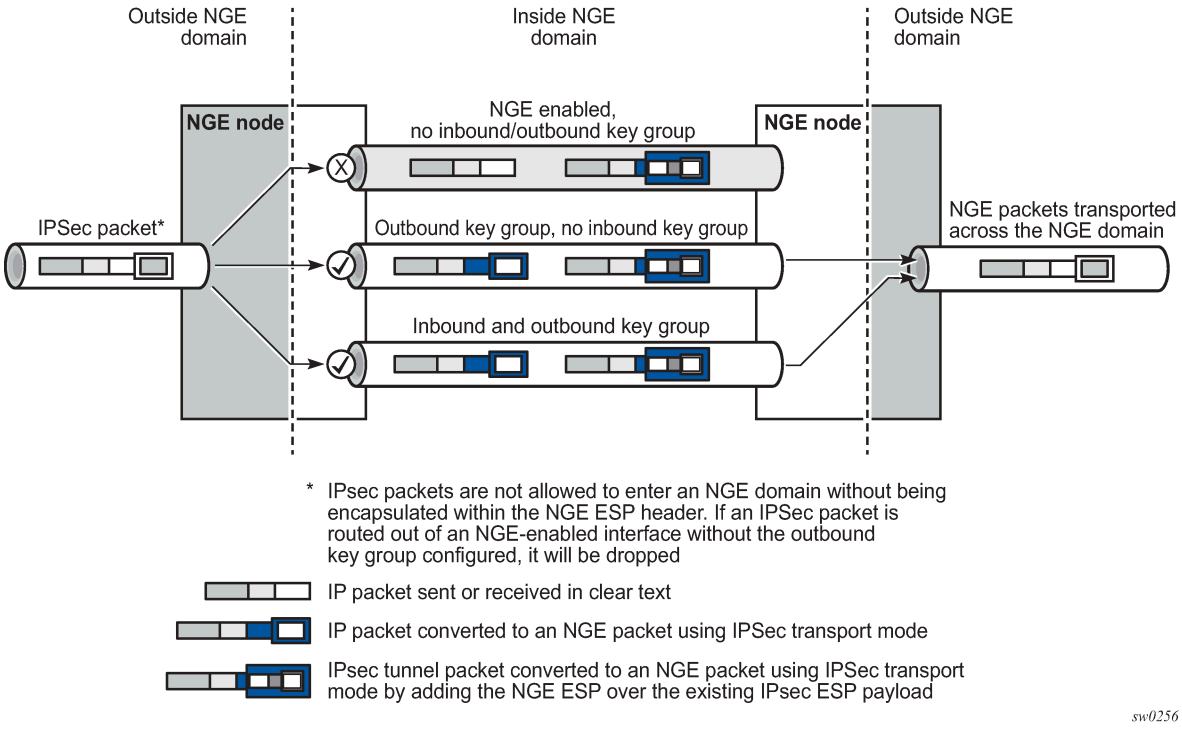
## 2.12.6 IPsec packets crossing an NGE domain

IPsec packets can cross the NGE domain because they are still considered Layer 3 packets. To avoid confusion between the security association used in an IPsec packet and the one used in a router interface NGE packet, the router always applies NGE to any IPsec packet that traverses the NGE domain.

IPsec packets that originate from a router within the NGE domain are not allowed to enter the NGE domain. The only exception to this restriction is OSPFv3 packets.

Figure 18: IPsec packets transiting an NGE domain shows how IPsec packets can transit an NGE domain.

*Figure 18: IPsec packets transiting an NGE domain*




An IPsec packet enters the router from outside the NGE domain. When the router determines that the egress interface to route the packet is inside an NGE domain, it selects an NGE router interface with one of the following configurations.

- NGE enabled with no inbound or outbound key group configured

  This link cannot forward the IPsec packet without adding the NGE ESP, but because nothing is configured for the outbound key group, the packet must be dropped.
- NGE enabled with outbound key group configured and no inbound key group configured — the packet originates outside the NGE domain, so the router adds an ESP header over the existing ESP and encrypts the payload using the NGE domain keys for the configured outbound key group.
- NGE enabled with both inbound and outbound key groups configured — the packet originates outside the NGE domain, so the router adds an ESP header over the existing ESP and encrypts the payload using the NGE domain keys for the configured outbound key group.

OSPFv3 IPsec support also uses IPsec transport mode packets. These packets originate from the CPM, which is considered outside the NGE domain; however, the above rules for encapsulating the packets with an NGE ESP apply and allow these packets to successfully transit the NGE domain.

## 2.12.7 Multicast packets traversing the NGE domain

Multicast packets that traverse an NGE domain can be categorized into two main scenarios:

- Scenario 1

  Multicast packets that ingress the router on an interface that is outside the NGE domain. These packets can egress a variety of interfaces that are either inside or outside the NGE domain.

- Scenario 2

  Multicast packets that ingress the router on an interface that is inside the NGE domain. These packets can egress a variety of interfaces that are either inside or outside the NGE domain. This scenario has two cases:

  – Scenario 2a

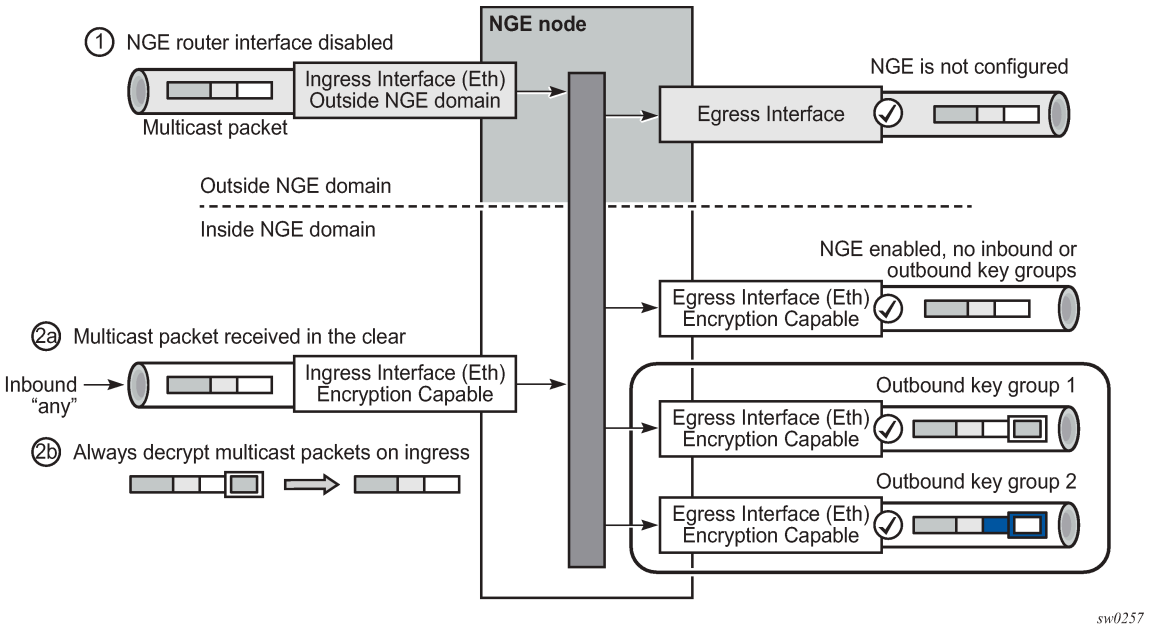
    The ingress multicast packet is not yet NGE-encrypted.

  – Scenario 2b

    The ingress multicast packet is NGE-encrypted.

Figure 19: Processing multicast packets shows these scenarios.

*Figure 19: Processing multicast packets*




Multicast packets received from outside the NGE domain (Scenario 1) are processed similarly to multicast packets received from inside the NGE domain (Scenarios 2a and 2b).

The processing rule is that multicast packets are always forwarded as clear text over the fabric. This means that for Scenario 2b, when a multicast packet is received on an encryption-capable interface and is NGE-encrypted, the packet is always decrypted first so that it can be processed in the same way as packets in Scenarios 1 and 2a.

On egress, the following scenarios apply:

- Egressing an interface outside the NGE domain

  Packets are processed in the same way as any multicast packets forwarded out a non-NGE interface.

- Egressing an NGE router interface and no inbound or outbound key group is configured

The router forwards these packets out from the egress interface without encrypting them because there is no outbound key group configured. This behavior also applies to unicast packets in the same scenario.

- egressing an NGE router interface with the outbound key group configured — the router encrypts the multicast packet using the SPI keys of the outgoing SA configured in the key group. This behavior also applies to unicast packets in the same scenario.

## 2.12.8 Assigning key groups to router interfaces

### Prerequisites

Assigning key groups to router interfaces involves the following three steps:

Step1 is required so that the router can initialize and differentiate the interface for NGE traffic before accepting or sending NGE packets. This assigns the interface to an NGE domain.

Assigning key groups to a router interface in steps 2 and 3 is similar to assigning key groups to SDPs or VPRN-based services. An outbound key group cannot be configured for a router interface without first enabling the **group-encryption** command.

When group-encryption is enabled and no inbound key group is configured, the router accepts NGE Layer 3 packets that were encrypted using keys from any security association configured in any key group on the system. If the packet specifies a security association that is not configured in any key group on the node, the packet is dropped.

The outbound key group references the key group to use when traffic egresses the router on the router interface. The inbound key group is used to make sure ingress traffic is using the correct key group on the router interface. If ingress traffic is not using the correct key group, the router counts these packets as errors.

### Procedure

**Step 1.** Enable NGE with the **group-encryption** command.

**Step 2.** Configure the outbound key group.

**Step 3.** Configure the inbound key group.

## 2.12.9 NGE and BFD support

When NGE is enabled on a router interface, BFD packets that originate from the network processor on the adapter card or from the system are encrypted in the same way as BFD packets that are generated by the CPM.

## 2.12.10 NGE and ACL interactions

When NGE is enabled on a router interface, the ACL function is applied as follows:

- **on ingress**

  Normal ACLs are applied to traffic received on the interface that could be either NGE-encrypted or clear text. For NGE-encrypted packets, this implies that only the source, destination, and IP options are available to filter on ingress, as the protocol is ESP, and the packet is encrypted. If an IP exception ACL

is also configured on the interface, the IP exception ACL is applied first to allow any clear text packets to ingress as needed. After the IP exception ACL is applied and if another filter or ACL is configured on the interface, the other filter processes the remaining packet stream (NGE-encrypted and IP exception ACL packets), and other ACL functions such as PBR or Layer 4 information filtering could be applied to any clear text packets that passed the exception ACL.

- **on egress**

  ACLs are applied to packets before they are NGE-encrypted as per normal operation without NGE enabled.

## 2.12.11 Router interface NGE and ICMP interactions over the NGE domain

Typically, ICMP works as expected over an NGE domain when all routers participating in the NGE domain are NGE-capable; this includes running an NGE domain over a private IP/MPLS network. When an ICMP message is required, the NGE packet is decrypted first, and the original packet is restored to create a detailed ICMP message using the original packet's header information.

When the NGE domain crosses a Layer 3 service provider, or crosses over routers that are not NGE-aware, it is not possible to create a detailed ICMP message using the original packet's information, as the NGE packet protocol is always set to ESP. Furthermore, the NGE router that receives these ICMP messages drops them because the messages are not NGE-encrypted.

The combination of dropping ICMP messages at the NGE border node and the missing unencrypted packet details in the ICMP information can cause problems with diagnosing network issues.

To help with diagnosing network issues, additional statistics are available on the interface to show whether ICMP messages are being returned from a foreign node. The following statistics are included in the group encryption NGE statistics for an interface:

- Group Enc Rx ICMP DestUnRch Pkts
- Group Enc Rx ICMP TimeExc Pkts
- Group Enc Rx ICMP Other Pkts

These statistics are used when clear text ICMP messages are received on an NGE router interface. The Invalid ESP statistics are not used in this situation even though the packet does not have a correct NGE ESP header. If there is no ingress exception ACL configured on the interface to allow the ICMP messages to be forwarded, the messages are counted and dropped.

If more information is required for these ICMP messages, such as source or destination address information, a second ICMP filter can be configured on the interface to allow logging of the ICMP messages. If the original packet information is also required, an egress exception ACL can be configured with the respective source or destination address information, or other criteria, to allow the original packet to enter the NGE domain in clear text and determine which flows are causing the ICMP failures.

## 2.12.12 1588v2 encryption with NGE

If a router interface is enabled for encryption and Layer 3 1588v2 packets are sent, they are encrypted using NGE. This means that if port timestamping is enabled on a router interface with NGE, the port timestamp is applied to the Layer 3 1588v2 packet using software-based timestamping instead of hardware-based timestamping, and consequently, timing accuracy may degrade. The exact level of timing or synchronization degradation is dependent on many factors, and testing is recommended to measure any impact.

If there is a need to support Layer 3 1588v2 with better accuracy for frequency or better time using port timestamping, an NGE exception ACL is required to keep the Layer 3 1588v2 packets in clear text. The exception ACL must enable UDP packets with destination port 319 to be sent in clear text.

## 2.13 Process overview

The following items are components to provision for basic router command options:

- **interface**

  A logical IP routing interface. When created, attributes like an IP address, port, link aggregation group, or the system can be associated with the IP interface.

- **address**

  The address associates the device's system name with the IP system address. An IP address must be assigned to each IP interface.

- **system interface**

  This creates an association between the logical IP interface and the system (loopback) address. The system interface address is the circuitless address (loopback) and is used by default as the router ID for protocols such as OSPF and BGP.

- **router ID**

  (Optional) The router ID specifies the router's IP address.

- **autonomous system**

  (Optional) An autonomous system (AS) is a collection of networks that are subdivided into smaller, more manageable areas.

- **confederation**

  (Optional) This option creates confederation-autonomous systems within an AS to reduce the number of IBGP sessions required within an AS.

## 2.14 Configuration notes

The following information describes router configuration requirements:

- A system interface and associated IP address must be specified.
- Boot options file (BOF) options must be configured before configuring router command options.
- Confederations can be configured before protocol connections (such as BGP) and peering command options are configured.
- IPv6 interfaces and associated routing protocols may only be configured on the following systems:
  - 7950 XRS systems
  - 7750 SR chassis systems
  - 7750 SR-a chassis systems
  - 7750 SR-e chassis systems
  - 7450 ESS systems with IPv6 functionality

## 2.15 Configuring an IP router with CLI

This section provides information to configure an IP router using CLI.

### 2.15.1 Router configuration overview

In a Nokia router, an interface is a logical named entity. An interface is created by specifying an interface name under the **configure router** context. This is the global router configuration context where objects like static routes are defined. An IP interface name can be up to 32 alphanumeric characters, must start with a letter, and is case-sensitive; for example, the interface name "1.1.1.1" is not allowed, but "int-1.1.1.1" is allowed.

If the interface name already exists, the router changes the context to maintain that IP interface. If the interface name already exists within another service ID or is an IP interface defined within the **configure router** commands, an error occurs, and the context does not change to that IP interface.

To create an interface, the following basic configuration tasks must be performed:

- Assign a name to the interface.
- Associate an IP address with the interface.
- Associate the interface with a network interface or the system interface.
- Configure appropriate routing protocols.

A system interface and network interface must be configured.

#### 2.15.1.1 System interface

The system interface is associated with a network entity (such as a specific Nokia router), not a specific interface. The system interface is also referred to as the loopback address. The system interface is associated during the configuration of the following entities:

- termination point of service tunnels
- hops when configuring MPLS paths and LSPs
- addresses on a target router for BGP and LDP peering

The system interface is used to preserve connectivity (when routing reconvergence is possible) when an interface fails or is removed. The system interface is used as the router identifier. A system interface must have an IP address with a 32-bit subnet mask.

#### 2.15.1.2 Network interface

A network interface can be configured on one of the following entities:

- physical or logical port
- SONET/SDH channel

For the 7950 XRS, a network interface can be configured on either a physical port or Ethernet LAG interface.

### 2.15.2 Basic configuration

See each specific chapter for specific routing protocol information and command syntax to configure protocols such as IS-IS and BGP.

The most basic router configuration must have the following:

- system name
- system address

The following example shows the 7750 SR and 7450 ESS router configuration.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
    autonomous-system 100
    router-id 10.10.10.103
    confederation {
        confed-as-num 1000
        members 100 { }
        members 200 { }
        members 300 { }
    }
...
    interface "system" {
        ipv4 {
            primary {
                address 10.10.10.103
                prefix-length 32
            }
        }
    }
    interface "to-104" {
        port 1/1/1
        ipv4 {
            primary {
                address 10.0.0.103
                prefix-length 24
            }
        }
    }
...
    isis 0 {
        loopfree-alternate {
        }
    }
```

**Example: classic CLI**

```
A:node-2>config# info
. . .
#------------------------------------------
# Router Configuration
#------------------------------------------
    router
        interface "system"
            address 10.10.10.103/32
        exit
        interface "to-104"
            address 10.0.0.103/24
```

```
            port 1/1/1
            exit
        exit
        autonomous-system 100
        confederation 1000 members 100 200 300
   router-id 10.10.10.103
...
    exit
    isis
    exit
...
#---------------------------------------
```

## 2.15.3 Common configuration tasks

The following sections describe basic system tasks.

### 2.15.3.1 Configuring a system name

Use the system command to configure a name for the device. The name is used in the prompt string. Only one system name can be configured. If multiple system names are configured, the last one configured overwrites the previous entry.

If special characters are included in the system name string, such as spaces, #, or ?, the entire string must be enclosed in double quotes.

**Example: MD-CLI**

```
[ex:/configure system]
A:admin@node-2# info
    name "node-2"
    location "Mt.View, CA, NE corner of FERG 1 Building"
    coordinates "37.390, -122.05500 degrees lat."
```

**Example: classic CLI**

```
A:node-2>config>system# info
#---------------------------------------
# System Configuration
#---------------------------------------
        name "node-2"
        location "Mt.View, CA, NE corner of FERG 1 Building"
        coordinates "37.390, -122.05500 degrees lat."
```

### 2.15.3.2 Configuring interfaces

The following command sequences create a system and a logical IP interface. The system interface assigns an IP address to the interface, then associates the IP interface with a physical port. The logical interface can associate attributes like an IP address or port.

The system interface cannot be deleted.

#### 2.15.3.2.1 Configuring a system interface

Use the following command to configure a system interface.

```
configure router interface
```

#### 2.15.3.2.2 Configuring a network interface

Use the commands in the following context to configure a network interface.

```
configure router interface
```

The following example shows network interface configuration information.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
...
interface "system" {
        ipv4 {
            primary {
                address 10.10.0.4/32
                prefix-length 32
            }
        }
     }
interface "to-ALA-2" {
        port 1/1/1
        egress {
            filter {
                ip "10"
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>router# info
#------------------------------------------
# IP Configuration
#------------------------------------------
        interface "system"
            address 10.10.0.4/32
        exit
        interface "to-ALA-2"
            address 10.10.24.4/24
            port 1/1/1
            egress
                filter ip 10
            exit
        exit
...
#------------------------------------------
```

Use the following command to enable CPU protection.

```
configure router interface cpu-protection
```

Use the commands in the following context to configure CPU protection policies.

```
configure system security cpu-protection
```

For more information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR System Management Guide*.

### 2.15.3.2.3 Assigning a key group to a router interface

**Note:** This implementation applies to the classic CLI.

The following example shows key group configuration for a router interface.

**Example: classic CLI**

```
A:node-2>config>router# info
----------------------------------------------
...
        interface demo
            group-encryption
                encryption-keygroup 6 direction inbound
                encryption-keygroup 6 direction outbound
                exit
            no shutdown
            exit
        exit
...
----------------------------------------------
```

### 2.15.3.2.4 Configuring IPv6

IPv6 interfaces and associated routing protocols may only be configured on the following systems:

- 7950 XRS systems
- 7750 SR chassis systems
- 7750 SR-a chassis systems
- 7750 SR-e chassis systems
- 7450 ESS chassis with IPv6 functionality

**Example: Default configuration when IPv6 is enabled on an interface (MD-CLI)**

```
[ex:/configure router "Base" interface "demo"]
A:admin@node-2# info
    admin-state enable
    port 1/2/37
    ipv6 {
        icmp6 {
            packet-too-big {
                number 100
```

```
                seconds 10
            }
            param-problem {
                number 100
                seconds 10
            }
            redirects {
                number 100
                seconds 10
            }
            time-exceeded {
                number 100
                seconds 10
            }
            unreachables {
                number 100
                seconds 10
            }
        }
    }
```

### Example: Default configuration when IPv6 is enabled on an interface (classic CLI)

```
A:node-2>config>router>if# info
----------------------------------------------
            port 1/2/37
            ipv6
            exit
            no shutdown

A:node-2>config>router>if>ipv6# info detail
----------------------------------------------
                icmp6
                    packet-too-big 100 10
                    param-problem 100 10
                    redirects 100 10
                    time-exceeded 100 10
                    unreachables 100 10
                exit
```

Use the commands in the following context to configure IPv6 on a router interface that you want to configure differently from the default configuration.

```
configure router interface ipv6 icmp6
```

### Example: Configuration of IPv6 on a router interface (MD-CLI)

```
[ex:/configure router "Base" interface "demo"]
A:admin@node-2# info
    port 1/2/3
    ipv4 {
        primary {
            address 10.11.10.1
            prefix-length 24
        }
    }
    ipv6 {
        address 2001:db8::1 {
            prefix-length 24
        }
```

```
    }
```

**Example: Configuration of IPv6 on a router interface (classic CLI)**

```
A:node-2>config>router>if# info
----------------------------------------------
            address 10.11.10.1/24
            port 1/2/37
            ipv6
                address 2001:db8::1/24
            exit
----------------------------------------------
```

### 2.15.3.2.5 Configuring IPv6 over IPv4

The following sections provide several examples of the features that must be configured (tunnel ingress and egress node) to implement IPv6 over IPv4 relay services for the 7750 SR OS.

### 2.15.3.2.6 Tunnel ingress node

The following example shows the configuration of the interface through which the IPv6 over IPv4 traffic leaves the node. This must be configured on a network interface.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
...
    static-routes {
        route 3ffe::c8c8:c802/128 route-type unicast {
            indirect 10.200.200.2 {
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>router# info
...
#------------------------------------------------
echo "Static Route Configuration"
#------------------------------------------------
        static-route-entry 3ffe::c8c8:c802/128
            indirect 10.200.200.2
                shutdown
                tunnel-next-hop
                    resolution disabled
                exit
            exit
        exit
----------------------------------------------
```

The following example shows the configuration of the network interface.

### Example: MD-CLI

```
[ex:/configure router "Base"]
A:admin@node-2# info
    interface "ip-1.1.1.1" {
        port 1/1/1
        ipv4 {
            primary {
                address 10.1.1.1
                prefix-length 30
            }
        }
    }
```

### Example: classic CLI

```
A:node-2>config>router# info
----------------------------------------------
...
        interface "ip-1.1.1.1"
            address 10.1.1.1/30
            port 1/1/1
          exit
...
----------------------------------------------
```

Both the IPv4 and IPv6 system addresses must be configured. The following example shows the configuration of interface information.

### Example: MD-CLI

```
[ex:/configure router "Base"]
A:admin@node-2# info
...
    interface "system" {
        ipv4 {
            primary {
                address 10.0.113.1
                prefix-length 32
            }
        }
        ipv6 {
            address 3ffe::c8c8:c801 {
                prefix-length 128
            }
        }
    }
```

### Example: classic CLI

```
A:node-2>config>router# info
----------------------------------------------
...
        interface "system"
            address 10.0.113.1/32
            ipv6
                address 3ffe::c8c8:c801/128
            exit
        exit
...
```

```
-----------------------------------------------
```

### 2.15.3.2.6.1 Learning the tunnel endpoint IPv4 system address

The following example shows the OSPF configuration to learn the IPv4 system address of the tunnel endpoint.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
    ospf 0 {
        area 0.0.0.0 {
            interface "ip-1.1.1.1" {
            }
            interface "system" {
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>router# info
-----------------------------------------------
...
        ospf
            area 0.0.0.0
                interface "system"
                exit
                interface "ip-1.1.1.1"
                exit
            exit
        exit
-----------------------------------------------
```

### 2.15.3.2.6.2 Configuring an IPv4 BGP peer

The following example shows the configuration of an IPv4 BGP peer with (IPv4 and) IPv6 protocol families.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
     bgp {
        router-id 203.0.113.1
        }
        export {
            policy ["ospf3"]
        }
        group "main" {
            type internal
            family {
                ipv4 true
                ipv6 true
            }
        }
        neighbor "203.0.113.2" {
```

```
            group "main"
            peer-as 1
            local-as {
                as-number 1
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>router# info
----------------------------------------------
...
        bgp
            export "ospf3"
            router-id 203.0.113.1
            group "main"
                family ipv4 ipv6
                type internal
                neighbor 203.0.113.2
                    local-as 1
                    peer-as 1
                exit
            exit
        exit
...
----------------------------------------------
```

### 2.15.3.2.6.3 IPv6 over IPv4 tunnel configuration example

The IPv6 address is the next-hop as it is received through BGP. The IPv4 address is the system address of the tunnel's endpoint.

See Configuring an IPv4 BGP peer for an example that shows the configuration of a policy to export IPv6 routes into BGP.

The following example shows an IPv6 over IPv4 tunnel configuration.

**Example: MD-CLI**

```
[ex:/configure policy-options]
A:admin@node-2# info
    policy-statement "ospf3" {
        description "Plcy Stmnt For 'From ospf3 To bgp'"
        entry 10 {
            description "Entry From Protocol ospf3 To bgp"
            from {
                protocol {
                    name [ospf3]
                }
            }
            to {
                protocol {
                    name [bgp]
                }
            }
            action {
                action-type accept
            }
        }
```

```
    }
```

**Example: classic CLI**

```
A:node-2>config>router# info
----------------------------------------------
...
        policy-options
            policy-statement "ospf3"
                description "Plcy Stmnt For 'From ospf3 To bgp'"
                entry 10
                    description "Entry From Protocol ospf3 To bgp"
                    from
                        protocol ospf3
                    exit
                    to
                        protocol bgp
                    exit
                    action accept
                    exit
                exit
            exit
        exit
...
----------------------------------------------
```

### 2.15.3.2.7 Tunnel egress node

The following example shows the configuration of the interface through which the IPv6 over IPv4 traffic leaves the node. It must be configured on a network interface. Both the IPv4 and IPv6 system addresses must be configured.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
...
  static-routes {
        route 3ffe::c8c8:c801/128 route-type unicast {
            indirect 10.0.113.1 {
                }
        }
```

**Example: classic CLI**

```
A:node-2>config>router# info
#--------------------------------------------------
"Static Route Configuration"
#--------------------------------------------------
        static-route-entry 3ffe::c8c8:c801/128
            indirect 10.0.113.1
...
                exit
            exit
        exit
```

The following example shows the network interface configuration with both IPv4 and IPv6 addresses configured.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
    interface "ip-1.1.1.2" {
        port 1/1/1
        ipv4 {
            primary {
                address 10.1.1.2
                prefix-length 30
            }
        }
    }
    interface "system" {
        ipv4 {
            primary {
                address 10.0.113.2
                prefix-length 32
            }
        }
        ipv6 {
            address 3ffe::c8c8:c802 {
                prefix-length 128
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>router# info
----------------------------------------------
...
        interface "ip-1.1.1.2"
            address 10.1.1.2/30
            port 1/1/1
        exit
        interface "system"
            address 10.0.113.2/32
            ipv6
                address 3ffe::c8c8:c802/128
            exit
        exit
----------------------------------------------
```

### 2.15.3.2.7.1 Learning the tunnel endpoint IPv4 system address

The following example shows the configuration of the OSPF configuration to learn the IPv4 system address of the tunnel endpoint.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
...
ospf 0 {
        area 0.0.0.0 {
            interface "ip-1.1.1.2" {
            }
            interface "system" {
```

```
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>router# info
----------------------------------------------
...
        ospf
            area 0.0.0.0
                interface "system"
                exit
                interface "ip-1.1.1.2"
                exit
            exit
        exit
----------------------------------------------
```

### 2.15.3.2.7.2 Configuring an IPv4 BGP peer

The following example shows the configuration an IPv4 BGP peer with (IPv4 and) IPv6 protocol families.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
...
    bgp {
        router-id 203.0.113.2
        export {
            policy ["ospf3"]
        }
        group "main" {
            type internal
            family {
                ipv4 true
                ipv6 true
            }
        }
        neighbor "203.0.113.1" {
            group "main"
            peer-as 1
            local-as {
                as-number 1
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2config>router# info
----------------------------------------------
...
        bgp
            export "ospf3"
            router-id 203.0.113.2
            group "main"
                family ipv4 ipv6
```

```
                type internal
                neighbor 203.0.113.1
                    local-as 1
                    peer-as 1
                exit
            exit
        exit
...
-------------------------------------------
```

### 2.15.3.2.7.3 IPv6 over IPv4 tunnel configuration example

The IPv6 address is the next-hop as it is received through BGP. The IPv4 address is the system address of the tunnel's endpoint.

See Configuring an IPv4 BGP peer for an example of the configuration of a policy to export IPv6 routes into BGP.

The following example shows an IPv6 over IPv4 tunnel configuration.

#### Example: MD-CLI

```
[ex:/configure policy-options]
A:admin@node-2# info
    policy-statement "ospf3" {
        description "Plcy Stmnt For 'From ospf3 To bgp'"
        entry 10 {
            description "Entry From Protocol ospf3 To bgp"
            from {
                protocol {
                    name [ospf3]
                }
            }
            to {
                protocol {
                    name [bgp]
                }
            }
            action {
                action-type accept
            }
        }
    }
```

#### Example: classic CLI

```
A:node-2>config>router# info
-------------------------------------------
...
        policy-options
            policy-statement "ospf3"
                description "Plcy Stmnt For 'From ospf3 To bgp'"
                entry 10
                    description "Entry From Protocol ospf3 To bgp"
                    from
                        protocol ospf3
                    exit
                    to
                        protocol bgp
                    exit
```

```
                    action accept
                    exit
                exit
            exit
        exit
-----------------------------------------------
```

### 2.15.3.2.8 Router advertisement

To configure the router to originate router advertisement messages on an interface, the interface must be configured under the router-advertisement context and be enabled. All other router advertisement configuration command options are optional.

Use the commands in the following contexts to configure router advertisement:

- **MD-CLI**

```
configure router ipv6 router-advertisement
configure service vprn ipv6 router-advertisement
```

- **classic CLI**

```
configure router router-advertisement
configure service vprn router-advertisement
```

The following example shows a router advertisement configuration.

#### Example: MD-CLI

```
[ex:/configure router "Base"]
A:admin@node-2# info
...
    ipv6 {
        router-advertisement {
            interface "n1" {
                admin-state enable
                use-virtual-mac true
                prefix 2001:db8:2::/64 {
                }
                prefix 2001:db8:3::/64 {
                    autonomous true
                    on-link true
                    preferred-lifetime 604800
                    valid-lifetime 2592000
                }
            }
        }
    }
```

#### Example: classic CLI

```
A:node-2>config>router>router-advert# info
-----------------------------------------------
            interface "n1"
                prefix 2001:db8:3::/64
                exit
                use-virtual-mac
                no shutdown
            exit
```

```
-----------------------------------------------
*A:node-2>config>router>router-advert# interface n1
*A:node-2>config>router>router-advert>if# prefix 2001:db8:3::/64
A:node-2>config>router>router-advert>if>prefix# info
-----------------------------------------------
                    autonomous
                    on-link
                    preferred-lifetime 604800
                    valid-lifetime 2592000
-----------------------------------------------
```

### 2.15.3.2.9 Configuring IPv6

The following example shows the default configuration when IPv6 is enabled on the interface.

**Example: Default IPv6 configuration (MD-CLI)**

```
[ex:/configure router "Base"]
A:admin@node-2# info
    interface "test" {
        port 1/3/37
        ipv6 {
        }
    }
[ex:/configure router "Base" interface "test" ipv6]
A:admin@node-2# info detail
...
    icmp6 {
        packet-too-big {
            admin-state enable
            number 100
            seconds 10
        }
        param-problem {
            admin-state enable
            number 100
            seconds 10
        }
        redirects {
            admin-state enable
            number 100
            seconds 10
        }
        time-exceeded {
            admin-state enable
            number 100
            seconds 10
        }
        unreachables {
            admin-state enable
            number 100
            seconds 10
        }
    }
```

**Example: Default IPv6 configuration (classic CLI)**

```
A:node-2>config>router# info
-----------------------------------------------
#-------------------------------------------------
```

```
   "IP Configuration"
#------------------------------------------------
        interface "test"
            port 1/3/37
            ipv6
                exit
            no shutdown
        exit
A:node-2>config>router>if>ipv6$ info detail
----------------------------------------------
                icmp6
                    packet-too-big 100 10
                    param-problem 100 10
                    redirects 100 10
                    time-exceeded 100 10
                    unreachables 100 10
                exit
```

The following example shows an IPv6 configuration.

**Example: IPv6 configuration (MD-CLI)**

```
[ex:/configure router "Base" interface "test"]
A:admin@node-2# info
    port 1/3/37
    ipv4 {
        primary {
            address 10.11.10.1
            prefix-length 24
        }
    }
    ipv6 {
        address 2001:db8::1 {
            prefix-length 24
        }
    }
```

**Example: IPv6 configuration (classic CLI)**

```
A:node-2>config>router>if# info
----------------------------------------------
            address 10.11.10.1/24
            port 1/3/37
            ipv6
                address 2001:db8::1/24
            exit
----------------------------------------------
```

### 2.15.3.2.9.1 IPv6 over IPv4 tunnel configuration example

The IPv6 address is the next-hop as it is received through BGP. The IPv4 address is the system address of the tunnel's endpoint.

Use the commands in the following context to export IPv6 routes into BGP.

```
configure router bgp export
```

The following example shows an IPv6 over IPv4 tunnel configuration.

**Example: MD-CLI**

```
[ex:/configure policy-options]
A:admin@node-2# info
    policy-statement "ospf3" {
        description "Plcy Stmnt For 'From ospf3 To bgp'"
        entry 10 {
            description "Entry From Protocol ospf3 To bgp"
            from {
                protocol {
                    name [ospf3]
                }
            }
            to {
                protocol {
                    name [bgp]
                }
            }
            action {
                action-type accept
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>router# info
----------------------------------------------
...
        policy-options
            policy-statement "ospf3"
                description "Plcy Stmnt For 'From ospf3 To bgp'"
                entry 10
                    description "Entry From Protocol ospf3 To bgp"
                    from
                        protocol ospf3
                    exit
                    to
                        protocol bgp
                    exit
                    action accept
                    exit
                exit
            exit
        exit
----------------------------------------------
```

## 2.15.3.2.10 Configuring proxy ARP

To configure proxy ARP, you can configure:

- A prefix list. Use the commands in the following context to configure a prefix list:
  - **MD-CLI**

    ```
    configure policy-options prefix-list
    ```

  - **classic CLI**

    ```
    configure router policy-options prefix-list
    ```

- A route policy statement and apply the specified prefix list. Use the commands in the following context to configure a route policy statement and apply the specified prefix list:
  - **MD-CLI**

    ```
    configure policy-options policy-statement
    ```

  - **classic CLI**

    ```
    configure router policy-options policy-statement
    ```

    - In the policy statement **entry to** context, specify the host source address(es) for which ARP requests can or cannot be forwarded to non-local networks, depending on the specified action.
    - In the policy statement **entry from** context, specify network prefixes that ARP requests to be forwarded or not forwarded depending on the action if a match is found. For more information about route policies, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide*.
- Apply the policy statement to the proxy ARP configuration. Use the commands in the following context to apply the policy statement to the proxy ARP configuration.

  ```
  configure router interface
  ```

### Example: Prefix list and policy statement configuration (MD-CLI)

```
[ex:/configure]
A:admin@node-2# info
 policy-options {
        prefix-list "prefixlist1" {
            prefix 10.20.30.0/24 type through {
                through-length 32
            }
        }
        policy-statement "ProxyARPpolicy" {
            entry 10 {
                from {
                    prefix-list ["prefixlist1"]
                }
                to {
                    prefix-list ["prefixlist2"]
                }
                action {
                    action-type reject
                }
            }
            default-action {
                action-type accept
            }
        }
    }
```

**Example: Prefix list and policy statement configuration (classic CLI)**

```
A:node-2>config>router>policy-options# info
----------------------------------------------
            prefix-list "prefixlist1"
                    prefix 10.20.30.0/24 through 32
            exit
            prefix-list "prefixlist2"
                    prefix 10.10.10.0/24 through 32
            exit
...
            policy-statement "ProxyARPpolicy"
                entry 10
                    from
                        prefix-list "prefixlist1"
                    exit
                    to
                        prefix-list "prefixlist2"
                    exit
                    action reject
                exit
                default-action accept
                exit
            exit
...
----------------------------------------------
```

The following example shows a proxy ARP configuration.

**Example: Proxy ARP configuration (MD-CLI)**

```
[ex:/configure router "Base" interface "iparptest"]
A:admin@node-2# info
    ipv4 {
        primary {
            address 192.0.2.59
            prefix-length 24
        }
        neighbor-discovery {
            local-proxy-arp true
            proxy-arp-policy ["ProxyARPpolicy"]
        }
    }
```

**Example: Proxy ARP configuration (classic CLI)**

```
A:node-2>config>router>if# info
----------------------------------------------
            address 192.0.2.59/24
            local-proxy-arp
            proxy-arp-policy "ProxyARPpolicy"
            exit
----------------------------------------------
```

### 2.15.3.3 Deriving the router ID

The router ID defaults to the address specified in the system interface command. If the system interface is not configured with an IP address, the router ID inherits the last four bytes of the MAC address.

Use the commands in the following context to configure the router ID manually.

```
configure router
```

Use the commands in the following context, on the BGP protocol level, to define a BGP router ID.

```
configure router bgp router-id
```

**Note:** A router ID configured under the **bgp router-id** context is only used within BGP.

If a new router ID is configured, protocols are not automatically restarted with the new router ID. The next time a protocol is initialized, the new router ID is used. An interim period of time can occur when different protocols use different router IDs. To force the new router ID, issue the shutdown and no shutdown commands for each protocol that uses the router ID, or restart the entire router.

It is possible to configure SR OS to operate with an IPv6 only BOF and no IPv4 system interface address. When configured in this manner, the user must explicitly define IPv4 router IDs for protocols such as OSPF and BGP because there is no mechanism to derive the router ID from an IPv6 system interface address.

The following example shows a router ID configuration.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
    router-id 10.10.0.4
    interface "system" {
        ipv4 {
            primary {
                address 10.10.0.4
                prefix-length 32
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2config>router# info
#------------------------------------------
# IP Configuration
#------------------------------------------
        interface "system"
            address 10.10.0.4/32
        exit
    . . .
        router-id 10.10.0.4
#------------------------------------------
```

### 2.15.3.4 Configuring a confederation

Configuring a confederation is optional. The AS and confederation topology design should be carefully planned. Autonomous system (AS), confederation, and BGP connection and peering must be explicitly created on each participating router. Identify AS numbers, confederation numbers, and members participating in the confederation.

See the BGP section for CLI syntax and command descriptions.

The following example shows the configuration of the confederation topology in Figure 2: Confederation configuration.

**Note:**

- Confederations can be preconfigured before configuring BGP connections and peering.
- Each confederation can have up to 15 members.

### Example: MD-CLI

```
[ex:/configure router "Base"]
A:admin@node-2# info
    autonomous-system 100
    router-id 10.10.10.103
    confederation {
        confed-as-num 2002
        members 200 { }
        members 300 { }
        members 400 { }
    }
    interface "system" {
        ipv4 {
            primary {
                address 10.10.10.103
                prefix-length 32
            }
        }
    }
    interface "to-104" {
        port 1/1/1
        ipv4 {
            primary {
                address 10.0.0.103
                prefix-length 24
            }
        }
    }
```

### Example: classic CLI

```
A:node-2>config>router# info
#------------------------------------------
# IP Configuration
#------------------------------------------
        interface "system"
            address 10.10.10.103/32
        exit
        interface "to-104"
            address 10.0.0.103/24
            port 1/1/1
        exit
        autonomous-system 100
        confederation 2002 members 200 300 400
        router-id 10.10.10.103

#------------------------------------------
```

### 2.15.3.5 Configuring an autonomous system

Configuring an autonomous system is optional. The following example shows the configuration of an autonomous system.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
    autonomous-system 100
    router-id 10.10.10.103
    interface "system" {
        ipv4 {
            primary {
                address 10.10.10.103
                prefix-length 32
            }
        }
    }
    interface "to-104" {
        port 1/1/1
        ipv4 {
            primary {
                address 10.0.0.103
                prefix-length 24
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>router# info
#------------------------------------------
# IP Configuration
#------------------------------------------
        interface "system"
            address 10.10.10.103/32
        exit
   interface "to-104"
            address 10.0.0.103/24
            port 1/1/1
            exit
        exit
        autonomous-system 100
        router-id 10.10.10.103
#------------------------------------------
```

### 2.15.3.6 Configuring overload state on a single SFM

When a router has fewer than the full set of SFMs functioning, the forwarding capacity can be reduced. Some scenarios include:

- fewer than the maximum number of SFMs installed in the system
- one or more SFMs have failed
- the system is in the ISSU process and the SFM is co-located on the CPM

An overload condition can be set for IS-IS and OSPF to enable the router to still participate in exchanging routing information, but route all traffic away from it when insufficient SFMs are active. Use the following commands to configure this overload condition:

- **MD-CLI**

```
configure router sfm-overload holdoff-time
configure service vprn sfm-overload holdoff-time
tools perform redundancy forced-single-sfm-overload
```

- **classic CLI**

```
configure router single-sfm-overload holdoff-time
configure service vprn single-sfm-overload holdoff-time
tools perform redundancy forced-single-sfm-overload
```

These cause an overload state in the IGP to trigger the traffic reroute by setting the overload bit in IS-IS or setting the metric to maximum in OSPF. When PIM uses IS-IS or OSPF to find out the upstream router, a next-hop change in the IS-IS or OSPF causes PIM to join the new path and prune the old path, which effectively also reroutes the multicast traffic downstream as well as the unicast traffic.

When the problem is resolved, and the required compliment of SFMs become active in the router, the overload condition is cleared, which causes the traffic to be routed back to the router.

The conditions to set overload are:

- 7750 SR-12/SR-7 and 7450 ESS-12/ESS-7 platforms: protocol sets overload if one of the SF/CPMs fails
- 7750 SR-12e and 7950 XRS platforms: protocol sets overload if two SFMs fail (two SFMs belonging to different SFM pairs on the XRS-40)

## 2.16 Service management tasks

This section describes IP router service management tasks.

### 2.16.1 Changing the system name

The system command sets the name of the device and is used in the prompt string. Only one system name can be configured. If multiple system names are configured, the last one configured overwrites the previous entry.

Use the following command to change the system name.

```
configure system name
```

The following example shows the system name change.

**Example: MD-CLI**

```
[ex:/configure system]
A:admin@node-2# info
    name "node-2"
    location "Mt.View, CA, NE corner of FERB 1 Building"
    coordinates "37.390, -122.05500 degrees lat."
```

```
    management-interface {
        configuration-mode mixed
    }
```

**Example: classic CLI**

```
A:node-2>config>system# info
#------------------------------------------------
echo "System Configuration"
#------------------------------------------------
        name "node-2"
        location "Mt.View, CA, NE corner of FERB 1 Building"
        coordinates "37.390, -122.05500 degrees lat."
        management-interface
            configuration-mode mixed
        exit
...
#------------------------------------------------
```

## 2.16.2 Modifying an interface configuration

This section provides examples of commands to use to modify the router interface configuration.

**Example: Modifying IP address information (MD-CLI)**

```
*[ex:/configure router "Base"]
A:admin@node-2# interface "to-sr1"

*[ex:/configure router "Base" interface "to-sr1"]
A:admin@node-2# admin-state disable

*[ex:/configure router "Base" interface "to-sr1"]
A:admin@node-2# ipv4

*[ex:/configure router "Base" interface "to-sr1" ipv4]
A:admin@node-2# primary

*[ex:/configure router "Base" interface "to-sr1" ipv4 primary]
A:admin@node-2# delete address

*[ex:/configure router "Base" interface "to-sr1" ipv4 primary]
A:admin@node-2# address 10.0.0.25

*[ex:/configure router "Base" interface "to-sr1" ipv4 primary]
A:admin@node-2# prefix-length 24

*[ex:/configure router "Base" interface "to-sr1" ipv4 primary]
A:admin@node-2# exit

*[ex:/configure router "Base" interface "to-sr1" ipv4]
A:admin@node-2# exit

*[ex:/configure router "Base" interface "to-sr1"]
A:admin@node-2# admin-state enable
```

**Example: Modifying the port information (MD-CLI)**

```
*[ex:/configure router "Base"]
A:admin@node-2# interface "to-sr1"
```

```
*[ex:/configure router "Base" interface "to-sr1"]
A:admin@node-2# admin-state disable

*[ex:/configure router "Base" interface "to-sr1"]
A:admin@node-2# delete port

*[ex:/configure router "Base" interface "to-sr1"]
A:admin@node-2# port 1/1/2

*[ex:/configure router "Base" interface "to-sr1"]
A:admin@node-2# admin-state enable
```

### Example: Modified output (MD-CLI)

```
[ex:/configure router "Base"]
A:admin@node-2# info
    interface "system" {
        ipv4 {
            primary {
                address 10.10.10.103
                prefix-length 32
            }
        }
    }
    interface "to-sr1" {
        admin-state enable
        port 1/1/2
        ipv4 {
            primary {
                address 10.0.0.25
                prefix-length 24
            }
        }
    }
```

### Example: Modifying IP address information (classic CLI)

```
*A:node-2>config>router# interface “to-sr1”
*A:node-2>config>router>if# shutdown
*A:node-2>config>router>if# no address
*A:node-2>config>router>if# address 10.0.0.25/24
*A:node-2>config>router>if# no shutdown
```

### Example: Modifying the port information (classic CLI)

```
*A:node-2>config>router# interface “to-sr1”
*A:node-2>config>router>if# shutdown
*A:node-2>config>router>if# no port
*A:node-2>config>router>if# port 1/1/2
*A:node-2>config>router>if# no shutdown
```

### Example: Modified output (classic CLI)

```
A:node-2>config>router# info
#------------------------------------------
# IP Configuration
#------------------------------------------
        interface "system"
            address 10.0.0.103/32
```

```
        exit
        interface "to-sr1"
            address 10.0.0.25/24
            port 1/1/2
        exit
        router-id 10.10.0.3
#------------------------------------------
```

### 2.16.3 Removing a key group from a router interface

**Note:** This implementation applies to the classic CLI.

The following example shows the commands to remove a key group from a router interface.

**Example: classic CLI**

```
*A:node-2>config>router# interface demo
*A:node-2>config>router>if# group-encryption
*A:node-2>config>router>if>group-encryp# no encryption-keygroup 6 direction inbound
*A:node-2>config>router>if>group-encryp# no encryption-keygroup 6 direction outbound
```

The following example shows that the key group configuration has been removed from a router interface.

**Example: classic CLI**

```
A:node-2>config>router# info
----------------------------------------------
...
        interface demo
            group-encryption
                exit
            no shutdown
            exit
        exit
...
----------------------------------------------
```

### 2.16.4 Changing the key group for a router interface

**Note:** This information applies to the classic CLI.

Use the following commands to change the key group on a router interface. The following example shows the inbound and outbound key groups being changed from key group 6 to key group 8.

**Example: classic CLI**

```
*A:node-2>config>router# interface demo
*A:node-2>config>router>if# group-encryption
*A:node-2>config>router>if>group-encryp# no encryption-keygroup 6 direction inbound
*A:node-2>config>router>if>group-encryp# encryption-keygroup 8 direction outbound
*A:node-2>config>router>if>group-encryp# encryption-keygroup 8 direction inbound
```

The following example shows that the key group configuration has been changed for the router interface.

### Example: classic CLI

```
A:node-2config>router# info
----------------------------------------------
...
        interface demo
            group-encryption
                encryption-keygroup 8 direction inbound
                encryption-keygroup 8 direction outbound
                exit
            no shutdown
            exit
        exit
...
----------------------------------------------
```

## 2.16.5 Deleting a logical IP interface

The following example shows how to delete a logical IP interface. Consider the following before you attempt to delete a logical IP interface:

1. Before an IP interface can be deleted, it must first be administratively disabled.
2. After the interface is administratively disabled, it can be deleted.

### Example: MD-CLI

In MD-CLI, the **delete** command used with the **interface** command removes the entry.

```
*[ex:/configure router "Base"]
A:admin@node-2# interface "test-interface"

*[ex:/configure router "Base"]
A:admin@node-2# admin-state disable

*[ex:/configure router "Base"]
A:admin@node-2# exit

*[ex:/configure router "Base"]
A:admin@node-2# delete interface "test-interface"
```

### Example: classic CLI

In classic CLI, the **no** form of the **interface** command typically removes the entry, but all entity associations must be shut down or deleted before an interface can be deleted.

```
*A:node-2>config>router# interface "test-interface"
*A:node-2>config>router>if$ shutdown
*A:node-2>config>router>if$ exit
*A:node-2>config>router# no interface "test-interface"
```

# 3 VRRP

## 3.1 VRRP overview

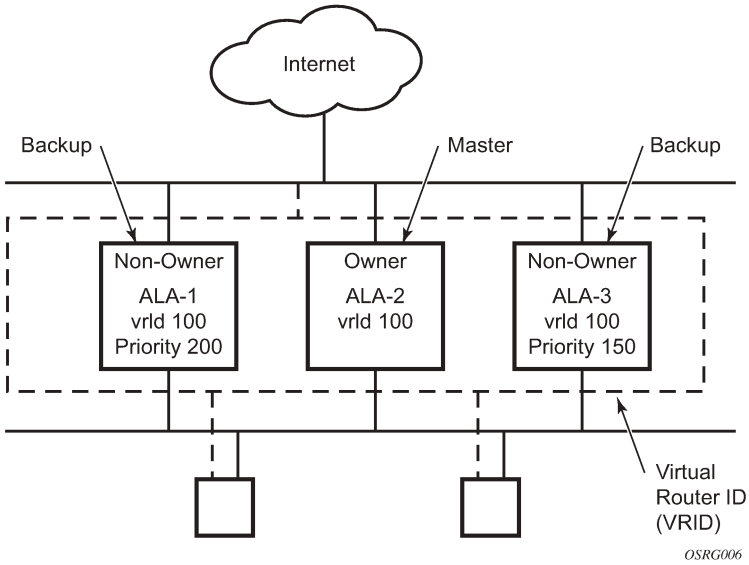
The Virtual Router Redundancy Protocol (VRRP) for IPv4 is defined in the IETF RFC 3768, *Virtual Router Redundancy Protocol*. VRRP for IPv6 is specified in *draft-ietf-vrrp-unified-spec-02.txt* and only applies to the 7750 SR and 7950 XRS. VRRP describes a method of implementing a redundant IP interface shared between two or more routers on a common LAN segment, allowing a group of routers to function as one virtual router. When this IP interface is specified as a default gateway on hosts directly attached to this LAN, the routers sharing the IP interface prevent a single point of failure by limiting access to this gateway address. VRRP can be implemented on IES service interfaces and on core network IP interfaces.

The VRRP standard RFC 3768 uses the term "master" state to denote the virtual router that is currently acting as the active forwarding router for the VRRP instance. This guide uses the term "active" as much as possible.

If the virtual router in master state fails, the backup router configured with the highest acceptable priority becomes the active virtual router. The new active router assumes the normal packet forwarding for the local hosts.

Figure 20: VRRP Configuration shows an example of a VRRP configuration.

*Figure 20: VRRP Configuration*




## 3.2 VRRP components

VRRP consists of the following components:

### 3.2.1 Virtual router

A virtual router is a logical entity managed by VRRP that acts as a default router for hosts on a shared LAN. It consists of a Virtual Router Identifier (VRID) and a set of associated IP addresses (or an address) across a common LAN. A VRRP router can be the backup for one or more virtual routers.

The purpose of supporting multiple IP addresses within a single virtual router is for multinetting. This is a common mechanism that allows multiple local subnet attachments on a single routing interface. Up to four virtual routers are possible on a single Nokia IP interface. The virtual routers must be in the same subnet. Each virtual router has its own VRID, state machine, and messaging instance.

### 3.2.2 IP address owner

VRRP can be configured in either an owner or non-owner mode. The owner is the VRRP router whose virtual router IP address is the same as the real interface IP address. This is the router that responds to packets addressed to one of the IP addresses for ICMP pings, TCP connections, and so on. All other virtual router instances participating in this message domain must have the same VRID configured and cannot be configured as owner.

Nokia routers allow the virtual routers to be configured as non-owners of the IP address. VRRP on a router can be configured to allow non-owners to respond to ICMP echo requests when they become the virtual router in master state for the VRRP instance. Telnet and other connection-oriented protocols can also be configured for master. However, the individual application conversations (connections) do not survive a VRRP failover. A non-owner VRRP router operating as a backup does not respond to any packets addressed to any of the virtual router IP addresses.

### 3.2.3 Primary and secondary IP addresses

A primary address is an IP address selected from the set of real interface address. VRRP advertisements are always sent using the primary IP address as the source of the IP packet.

An IP interface must always have a primary IP address assigned for VRRP to be active on the interface. Nokia routers supports both primary and secondary IP addresses (multinetting) on the IP interface. The virtual router's VRID primary IP address is always the primary address on the IP interface. VRRP uses the primary IP address as the IP address placed in the source IP address field of the IP header for all VRRP messages sent on that interface.

### 3.2.4 Virtual router

The VRRP router that controls the IP addresses associated with a virtual router is considered to be in the master state and is the active router for the VRRP instance and is responsible for forwarding packets sent to the VRRP IP address. The election process provides dynamic failover of the forwarding responsibility if the master becomes unavailable. This allows any of the virtual router IP addresses on the LAN to be used as the default first hop router by end hosts. This enables a higher availability default path without requiring configuration of dynamic routing or router discovery protocols on every end host.

If the master is unavailable, each backup virtual router for the VRID compares the configured priority values to determine the master role. In case of a tie, the virtual router with the highest primary IP address becomes master.

The preempt command option can be set to false to prevent a backup virtual router with a better priority value from becoming master when an existing non-owner virtual router is the current master. This is determined on a first-come, first-served basis.

While master, a virtual router routes and originates all IP packets into the LAN using the physical MAC address for the IP interface as the Layer 2 source MAC address, not the VRID MAC address. ARP packets also use the parent IP interface MAC address as the Layer 2 source MAC address while inserting the virtual router MAC address in the appropriate hardware address field. VRRP messages are the only packets transmitted using the virtual router MAC address as the Layer 2 source MAC address.

### 3.2.5 Virtual router backup

A new virtual router master is selected from the set of VRRP routers available to assume forwarding responsibility for a virtual router in case the current master fails.

### 3.2.6 Owner and non-owner VRRP

The owner controls the IP address of the virtual router and is responsible for forwarding packets sent to this IP address. The owner assumes the role of the master virtual router. Only one virtual router in the domain can be configured as owner. All other virtual router instances participating in this message domain must have the same VRID configured.

The most important command option to be defined on a non-owner virtual router instance is the priority. The priority defines a virtual router's selection order in the master election process. The priority value and the preempt mode determine the virtual router with the highest priority to become the master virtual router.

The base priority is used to determine the in-use priority of the virtual router instance as modified by any optional VRRP priority control policy. VRRP priority control policies can be used to either override or adjust the base priority value depending on events or conditions within the chassis.

For information about non-owner access command options, see VRRP non-owner accessibility.

For owner virtual router instances, use the following commands to define the IP addresses that are advertised within VRRP advertisement messages:

- **MD-CLI**

```
configure router interface ipv4 vrrp backup
configure router interface ipv6 vrrp backup
```

- **classic CLI**

```
configure router interface vrrp backup
configure router interface ipv6 vrrp backup
```

For owner virtual router instances, after you define the IP addresses that are advertised within VRRP advertisement messages, this communicates the IP addresses that the master is advertising to backup virtual routers receiving the messages. The specified unicast IPv4 address must be equal to one of the existing IP addresses in the parental IP interface (primary or secondary) or the **backup** command fails.

For non-owner virtual router instances, the **backup** command for IPv4 or IPv6 creates an IP interface IP address used for routing IP packets and communicating with the system, based on which access command options are enabled (ntp-reply, ping-reply, telnet-reply, and ssh-reply). The specified unicast IPv4 address must exist on one of the local subnets of the parental IP interface. If the specified address does

not exist on one of the local subnets of the parental IP interface or if the specified address uses the same IP address as the parental IP interface, the **backup** command fails.

The **backup** command must be executed successfully at least once before the virtual router instance can enter the operational state.

The new interface IP address created with the **backup** command assumes the mask and command options of the corresponding parent IP interface IP address. The unicast IPv4 address is only active when the virtual router instance is operating in the master state. When not operating as master, the virtual router instance acts as if it is operationally down. It does not respond to ARP requests made to the unicast IPv4 address, nor does it route packets received with its VRID-derived source MAC address. A non-master virtual router instance always silently discards packets destined for the unicast IPv4 address. One virtual router instance may only have one virtual router IP address from a parental local subnet. Multiple virtual router instances can define a virtual router IP address from the same local subnet, as long as each IP address is different.

### 3.2.7 Configurable command options

As well as to backup IP addresses, to facilitate configuration of a virtual router on Nokia routers, the following command options can be defined in owner configurations:

- Virtual Router ID (VRID)
- Message interval and master inheritance
- VRRP message authentication
- Authentication Data
- Virtual MAC Address

The following command options can be defined in non-owner configurations:

- Virtual Router ID (VRID)
- Priority
- Message interval and master inheritance
- Master Down Interval
- Preempt Mode
- VRRP message authentication
- Authentication Data
- Virtual MAC Address
- Inherit Master VRRP Router’s Advertisement Interval Timer
- Policies

#### 3.2.7.1 Virtual Router ID (VRID)

The VRID must be configured with the same value on each virtual router associated with the redundant IP address (or addresses). The VRID is placed in all VRRP advertisement messages sent by each virtual router.

#### 3.2.7.2 Priority

The priority value affects the interaction between this VRID and the same VRID of other virtual routers participating on the same LAN. A higher-priority value defines a greater priority in becoming the virtual router master for the VRID. The priority value can only be configured when the defined IP address on the IP interface is different from the virtual router IP address (non-owner mode).

When the IP address on the IP interface matches the virtual router IP address (owner mode), the priority value is fixed at 255, the highest value possible. This virtual router member is considered the owner of the virtual router IP address. There can only be one owner of the virtual router IP address for all virtual router members.

The priority value 0 is reserved for VRRP advertisement message purposes. It is used to tell other virtual routers in the same VRID that this virtual router is no longer acting as master, triggering a new election process. When this happens, each backup virtual router sets its master down timer equal to the skew time value. This shortens the time until one of the backup virtual routers becomes master.

The current master virtual router must transmit a VRRP advertisement message immediately upon receipt of a VRRP message with priority set to 0. This prevents another backup from becoming master for a short period of time.

Non-owner virtual routers may be configured with a priority of 254 through 1. The default value is 100. Multiple non-owners can share the same priority value. When multiple non-owner backup virtual routers are tied (transmit VRRP advertisement messages simultaneously) in the election process, all attempts to become master simultaneously. The one with the best priority wins the election. If the priority value in the message is equal to the master’s local priority value, the primary IP address of the local master and of the message is evaluated as the tie breaker. The higher IP address becomes master. (The primary IP address is the source IP address of the VRRP advertisement message.)

The priority value is also used to determine when to preempt the existing master. If the preempt mode value is true, VRRP advertisement messages from inferior (lower-priority) masters are discarded, causing the master down timer to expire and causing the transition to master state.

The priority value also dictates the skew time added to the master timeout period.

#### 3.2.7.3 IP Addresses

Each virtual router with the same VRID should be defined with the same set of IP addresses. These are the IP addresses being used by hosts on the LAN as gateway addresses. Multinetting supports 16 IP addresses on the IP interface; up to 16 addresses can be assigned to a specific virtual router instance.

#### 3.2.7.4 Message interval and master inheritance

Each virtual router is configured with a message interval per VRID within which it participates. This command option must be the same for every virtual router on the VRID.

For IPv4, the default advertisement interval is 1 s and can be configured between 100 ms and 255 s 900 ms. For IPv6, the default advertisement interval is 1 s and can be configured between 100 ms and 40 s 950 ms.

As specified in the RFC, the advertisement interval field in every received VRRP advertisement message must match the locally configured advertisement interval. If a mismatch occurs, depending on the inherit configuration, the current master's advertisement interval setting can be used to operationally override the

locally configured advertisement interval setting. If the current master changes, the new master setting is used. If the local virtual router becomes master, the locally configured advertisement interval is enforced.

If a VRRP advertisement message is received with an advertisement interval set to a value different from the local value and the inherit command option is disabled, the message is discarded without processing.

The master virtual router on a VRID uses the advertisement interval to load the advertisement timer, specifying when to send the next VRRP advertisement message. Each backup virtual router on a VRID uses the advertisement interval (with the configured local priority) to determine the master down timer value.

VRRP advertisement messages that are fragmented, or contain IP options (IPv4), or contain extension headers (IPv6) require a longer message interval to be configured.

#### 3.2.7.5 Skew time

The skew time is used to add a time period to the master down interval. This is not a configurable command option. It is determined from the current local priority of the virtual router's VRID. To calculate the skew time, the virtual router evaluates the following formula:

For IPv4: Skew Time equals ((256 - priority) / 256) seconds

For IPv6: Skew Time equals (((256 - priority) * Master_Adver_Interval) / 256) centiseconds

The higher the priority value, the shorter the skew time is. This means that virtual routers with a lower priority transition to master slower than virtual routers with a higher priority.

#### 3.2.7.6 Master Down Interval

The master down interval is a calculated value used to load the master down timer. When the master down timer expires, the virtual router enters the master state. To calculate the master down interval, the virtual router evaluates the following formula:

The master down interval equals (3 x Operational Advertisement Interval) + Skew Time

The operational advertisement interval is dependent upon the state of the inherit command option. When the inherit command option is enabled, the operational advertisement interval is determined from the current master's advertisement interval field in the VRRP advertisement message. When inherit is disabled, the operational advertisement interval must be equal to the locally configured advertisement interval.

The master down timer is only operational when the local virtual router is operating in backup mode.

#### 3.2.7.7 Preempt Mode

Preempt mode is a true or false configured value that controls whether a specific backup virtual router preempts a lower-priority master. The IP address owner always becomes master when available. Preempt mode cannot be set to false on the owner virtual router. The default value for preempt mode is true.

When the preempt mode is true, a master non-owner virtual router only allows itself to be preempted when the incoming VRRP advertisement message priority field value is one of the following:

- Greater than the virtual router in-use priority value
- Equal to the in-use priority value, and the source IP address (primary IP address) is greater than the virtual router instance primary IP address

A backup router only attempts to become the master router if the preempt mode is true and the received VRRP advertisement priority field is less than the virtual router in-use priority value.

### 3.2.7.8 VRRP message authentication

The authentication type command option defines the type of authentication used by the virtual router in VRRP advertisement message authentication. VRRP message authentication is applicable to IPv4 only. The current master uses the configured authentication type to indicate any egress message manipulation that must be performed in conjunction with any supporting authentication command options before transmitting a VRRP advertisement message. The configured authentication type value is transmitted in the message authentication type field with the appropriate authentication data field filled in. Backup routers use the authentication type message field value in interpreting the contained authentication data field within received VRRP advertisement messages.

VRRP supports three message authentication methods that provide varying degrees of security. The supported authentication types are:

- 0 – No Authentication
- 1 – Simple Text Password
- 2 – IP Authentication Header

#### 3.2.7.8.1 Authentication Type 0 – No Authentication

The use of authentication type 0 indicates that VRRP advertisement messages are not authenticated (provides no authentication). The master transmitting VRRP advertisement messages transmits the value 0 in the egress messages authentication type field and the authentication data field. Backup virtual routers receiving VRRP advertisement messages with the authentication type field equal to 0 ignores the authentication data field in the message.

All compliant VRRP advertisement messages are accepted. The following fields within the received VRRP advertisement message are checked for compliance (the VRRP specification may require additional checks):

- IP header checks specific to VRRP
  - IP header destination IP address – must be 224.0.0.18
  - IP header TTL field – must be equal to 255; the packet must not have traversed any IP routed hops
  - IP header protocol field – must be 112 (decimal)
- VRRP message checks
  - Version field – must be set to the value of 2
  - Type field – must be set to the value of 1 (advertisement)
  - Virtual router ID field – must match one of the configured VRIDs on the ingress IP interface (all other fields are dependent on matching the virtual router ID field to one of the interfaces configured VRID command options)
  - Priority field – must be equal to or greater than the VRID in-use priority or be equal to 0 (if equal to the VRID in-use priority and 0, requires further processing about master/backup and senders IP address to determine validity of the message)
  - Authentication type field – must be equal to 0

– Advertisement interval field – must be equal to the VRID configured advertisement interval
– Checksum field – must be valid
– Authentication data fields – must be ignored

VRRP messages not meeting the criteria are silently discarded.

#### 3.2.7.8.2 Authentication Type 1 – Simple Text Password

The use of authentication type 1 indicates that VRRP advertisement messages are authenticated with a clear (simple) text password. All virtual routers participating in the virtual router instance must be configured with the same 8 octet password. Transmitting virtual routers put a value of 1 in the VRRP advertisement message authentication type field and put the configured simple text password into the message authentication data field. Receiving virtual routers compare the message authentication data field with the local configured simple text password based on the message authentication type field value of 1.

The same checks are performed as for type 0, with the following exceptions (the VRRP specification may require additional checks):

VRRP message checks

- Authentication type field – must be equal to 1
- Authentication data fields – must be equal to the VRID configured simple text password

Any VRRP messages not meeting the type 0 verification checks, with the preceding exceptions are silently discarded.

#### 3.2.7.8.3 Authentication Failure

Any received VRRP advertisement message that fails authentication must be silently discarded with an invalid authentication counter incremented for the ingress virtual router instance.

### 3.2.7.9 Authentication Data

This feature is different from the VRRP advertisement message field with the same name. Authentication data is any required authentication information that is pertinent to the configured authentication type. The type of authentication data used for each authentication type is listed in Table 5: Authentication Data Type.

*Table 5: Authentication Data Type*

| Authentication Type | Authentication Data |
|---|---|
| 0 | None, authentication is not performed |
| 1 | Simple text password consisting of 8 octets |

### 3.2.7.10 Virtual MAC Address

The MAC address can be used instead of an IP address in ARP responses when the virtual router instance is master. The MAC address configuration must be the same for all virtual routers participating as a virtual router, or indeterminate connectivity by the attached IP hosts results. All VRRP advertisement messages are transmitted with *ieee-mac-address* as the source MAC.

### 3.2.7.11 VRRP Advertisement Message IP Address List Verification

VRRP advertisement messages contain an IP address count field that indicates the number of IP addresses listed in the sequential IP address fields at the end of the message.

The Nokia routers implementation always logs mismatching events. The decision on where and whether to forward the generated messages depends on the configuration of the event manager.

To facilitate the sending of mismatch log messages, each virtual router instance keeps the mismatch state associated with each source IP address in the VRRP master table. Whenever the state changes, a mismatch log message is generated indicating the source IP address within the message, the mismatch or match event, and the time of the event.

With secondary IP address support, multiple IP addresses can be in the list and each should match the IP address on the virtual router instance. Owner and non-owner virtual router instances have the supported IP addresses explicitly defined, making mismatched supported IP addresses within the interconnected virtual router instances a provisioning issue.

### 3.2.7.12 Inherit Master VRRP Router’s Advertisement Interval Timer

The virtual router instance can inherit the master VRRP router’s advertisement interval timer, which is used by backup routers to calculate the master down timer.

The inheritance is only configurable in the non-owner nodal context. The inheritance is used to allow the current virtual router instance master to dictate the master down timer for all backup virtual routers.

### 3.2.7.13 IPv6 Virtual Router Instance Operationally Up

After the 7750 SR or 7950 XRS IPv6 virtual router is configured with a minimum of one link-local backup address, the parent interface’s router advertisement must be configured to use the virtual MAC address for the virtual router to be considered operationally up.

### 3.2.7.14 Policies

Policies can be configured to control VRRP priority with the virtual router instance. VRRP priority control policies can be used to override or adjust the base priority value, depending on events or conditions within the chassis.

The policy can be associated with more than one virtual router instance. The priority events within the policy override or diminish the base priority dynamically affecting the in-use priority. As priority events clear in the policy, the in-use priority can eventually be restored to the base priority value.

Policies can only be configured in the non-owner VRRP context. For non-owner virtual router instances, if policies are not configured, then the base priority is used as the in-use priority.

## 3.3 VRRP priority control policies

This implementation of VRRP supports control policies to manipulate virtual router participation in the VRRP master election process and master self-deprecation. The local priority value for the virtual router instance is used to control the election process and master state.

### 3.3.1 VRRP virtual router policy constraints

Priority control policies can only be applied to non-owner VRRP virtual router instances. Owner VRRP virtual routers cannot be controlled by a priority control policy because they are required to have a priority value of 255 that cannot be diminished. Only one VRRP priority control policy can be applied to a non-owner virtual router instance.

Multiple VRRP virtual router instances may be associated with the same IP interface, allowing multiple priority control policies to be associated with the IP interface.

An applied VRRP priority control policy only affects the in-use priority on the virtual router instance when the preempt mode has been enabled. A virtual router instance with preempt mode disabled always uses the base priority as the in-use priority, ignoring any configured priority control policy.

### 3.3.2 VRRP virtual router instance base priority

Non-owner virtual router instances must have a base priority value between 1 and 254. The value 0 is reserved for master termination. The value 255 is reserved for owners. The default base priority for non-owner virtual router instances is the value 100.

The base priority is the starting priority for the VRRP instance. The actual in-use priority for the VRRP instance is determined from the base priority and an optional VRRP priority control policy.

### 3.3.3 VRRP priority control policy delta in-use priority limit

A VRRP priority control policy enforces an overall minimum value that the policy can assign to the VRRP virtual router instance base priority. This value provides a lower limit to the delta priority events manipulation of the base priority.

A delta priority event is a conditional event defined in the priority control policy that subtracts a specified amount from the current, in-use priority for all VRRP virtual router instances to which the policy is applied. Multiple delta priority events can apply simultaneously, creating a dynamic priority value. The base priority for the instance, less the sum of the delta values, determines the priority value in-use.

An explicit priority event is a conditional event defined in the priority control policy that explicitly defines the in-use priority value for the virtual router instance. The explicitly defined value is not affected by the delta in-use priority limit. When multiple explicit priority events happen simultaneously, the lowest value is used for the in-use priority. The configured base priority is not a factor in explicit priority overrides of the in-use priority.

The allowed range of the Delta In-Use Priority Limit is 1 to 254. The default is 1, which prevents the delta priority events from operationally disabling the virtual router instance.

### 3.3.4 VRRP priority control policy priority events

The main function of a VRRP priority control policy is to define conditions or events that affect the system's ability to communicate with outside hosts or portions of the network. When one or multiple of these events are true, the base priority on the virtual router instance is either overwritten with an explicit value, or a sum of delta priorities is subtracted from the base priority. The result is the in-use priority for the virtual router instance. Any priority event may be configured as an explicit event or a delta event.

Explicit events override all delta events. When multiple explicit events occur, the event with the lowest priority value is assigned to the in-use priority. As events clear, the in-use priority is reevaluated accordingly and adjusted dynamically.

Delta priority events also have priority values. When no explicit events have occurred within the policy, the sum of the occurring delta events priorities is subtracted from the base priority of each virtual router instance. If the result is lower than the delta in-use priority limit, the delta in-use priority limit is used as the in-use priority for the virtual router instance. Otherwise, the in-use priority is set to the base priority less the sum of the delta events.

Each event generates a VRRP priority event message indicating the policy-id, the event type, the priority type (delta or explicit), and the event priority value. Another log message is generated when the event is no longer true, indicating that it has been cleared.

#### 3.3.4.1 Priority event hold-set timers

Hold-set timers are used to dampen the effect of a flapping event. A flapping event is where the event continually transitions between clear and set. The hold-set value is loaded into a hold-set timer that prevents a set event from transitioning to the cleared state until it expires.

Each time an event transitions between cleared and set, the timer is loaded and begins to count down to zero. If the timer reaches zero, the event is allowed to enter the cleared state again. Entering the cleared state is always dependent on the object controlling the event conforming to the requirements defined in the event. It is possible, on some event types, to have a further set action reload the hold-set timer. This extends the time that must pass before the hold-set timer expires, and the event enters the cleared state.

For an example of a hold-set timer setting, see LAG degrade priority event.

#### 3.3.4.2 Port down priority event

The port down priority event is assigned to either a physical port or a SONET/SDH channel for the 7750 SR and 7450 ESS. The port or channel operational state is evaluated to determine a port down priority event or event clear.

When the port or channel operational state is up, the port down priority event is considered false or cleared. When the port or channel operational state is down, the port down priority event is considered true or set.

### 3.3.4.3 LAG degrade priority event

The LAG degrade priority event is assigned to an existing Link Aggregation Group (LAG). The LAG degrade priority event is conditional on a percentage of available port bandwidth on the LAG. Multiple bandwidth percentage thresholds may be defined, each with its own priority value.

If the LAG transitions from one threshold to the next, the previous threshold priority value is subtracted from the total delta sum while the new threshold priority value is added to the sum. The new sum is then subtracted from the base priority and compared to the delta in-use priority limit to determine the new in-use priority on the virtual router instance.

The following example shows a LAG degrade priority event and its interaction with the hold-set timer in changing the in-use priority.

The following state and timer settings are used for the LAG events listed in Table 6: LAG events:

- User-defined thresholds: 2 ports down, 4 ports down, 6 ports down
- LAG configured ports: 8 ports
- Hold-set timer (hold-set): 5 seconds

*Table 6: LAG events*

| Time (seconds) | LAG port state | Command option | State | Comments |
|---|---|---|---|---|
| 0 | All ports down | Event State | Set - 8 ports down | — |
| | | Event Threshold | 6 ports down | — |
| | | Hold-set Timer | 5 seconds | Set to **hold-set** command option |
| 1 | One port up | Event State | Set - 8 ports down | Cannot change until hold-set timer expires |
| | | Event Threshold | 6 ports down | — |
| | | Hold-set Timer | 5 seconds | Event does not affect timer |
| 2 | All ports up | Event State | Set - 8 ports down | Still waiting for hold-set timer expiry |
| | | Event Threshold | 6 ports down | — |
| | | Hold-set Timer | 3 seconds | — |
| 5 | All ports up | Event State | Cleared - All ports up | — |
| | | Event Threshold | None | Event cleared |
| | | Hold-set Timer | Expired | — |
| 100 | Five ports down | Event State | Set - 5 ports down | — |
| | | Event Threshold | 4 ports down | — |

| Time (seconds) | LAG port state | Command option | State | Comments |
|---|---|---|---|---|
| | | Hold-set Timer | Expired | Set to **hold-set** command option |
| 102 | Three ports down | Event State | Set - 5 ports down | — |
| | | Event Threshold | 4 ports down | — |
| | | Hold-set Timer | 3 seconds | — |
| 103 | All ports up | Event State | Set - 5 ports down | — |
| | | Event Threshold | 4 ports down | — |
| | | Hold-set Timer | 2 second | — |
| 104 | Two ports down | Event State | Set - 5 ports down | — |
| | | Event Threshold | 4 ports down | — |
| | | Hold-set timer | 1 second | Current threshold is 5, so 2 down has no effect |
| 105 | Two ports down | Event State | Set - 2 ports down | — |
| | | Event Threshold | 2 ports down | — |
| | | Hold-set timer | Expired | — |
| 200 | Four ports down | Event State | Set - 2 ports down | — |
| | | Event Threshold | 4 ports down | — |
| | | Hold-set timer | 5 seconds | Set to hold-set command option |
| 202 | Seven ports down | Event State | Set - 7 ports down | Changed because of increase |
| | | Event Threshold | 6 ports down | — |
| | | Hold-set timer | 5 seconds | Set to **hold-set** because of threshold increase |
| 206 | All ports up | Event State | Set - 7 ports down | — |
| | | Event Threshold | 6 ports down | — |
| | | Hold-set timer | 1 second | — |
| 207 | All ports up | Event State | Cleared - All ports up | — |
| | | Event Threshold | None | Event cleared |
| | | Hold-set timer | Expired | — |

#### 3.3.4.4 Host unreachable priority event

The host unreachable priority event creates a continuous ping task that is used to test connectivity to a remote host. The path to the remote host and the remote host itself must be capable and configured to accept ICMP echo request and replies for the ping to be successful.

The ping task is controlled by interval and size parameters that define how often the ICMP request messages are transmitted and the size of each message. A historical missing reply command option defines when the ping destination is considered unreachable.

When the host is unreachable, the host unreachable priority event is considered true or set. When the host is reachable, the host unreachable priority event is considered false or cleared.

#### 3.3.4.5 Route unknown priority event

The route unknown priority event defines a task that monitors the existence of a specific route prefix in the system's routing table.

The route monitoring task can be constrained by a condition that allows a prefix, which is less specific than the defined prefix, to be considered as a match. The source protocol can be defined to indicate the protocol that the installed route must be populated from. To further define match criteria when multiple instances of the route prefix exist, an optional next hop command option can be defined.

When a route prefix exists within the active route table that matches the defined match criteria, the route unknown priority event is considered false or cleared. When a route prefix does not exist within the active route table matching the defined criteria, the route unknown priority event is considered true or set.

## 3.4 VRRP non-owner accessibility

Although the RFC states that only VRRP owners can respond to ping and other management-oriented protocols directed to the VRID IP addresses, the routers allow an override of this restraint on a per VRRP virtual router instance basis.

### 3.4.1 Non-owner access ping reply

When non-owner access ping reply is enabled on a virtual router instance, ICMP echo request messages destined for the non-owner virtual router instance IP addresses are not discarded at the IP interface when operating in master mode. ICMP echo request messages are always discarded in backup mode.

When non-owner access ping reply is disabled on a virtual router instance, ICMP echo request messages destined for the non-owner virtual router instance IP addresses are silently discarded in both the master and backup modes.

### 3.4.2 Non-owner access Telnet

When non-owner access Telnet is enabled on a virtual router instance, authorized Telnet sessions may be established that are destined for the virtual router instance IP addresses when operating in master mode. Telnet sessions are always discarded at the IP interface when destined for a virtual router IP address operating in backup mode. Enabling non-owner access Telnet does not guarantee Telnet access; correct

management and security features must be enabled to allow Telnet on this interface and possibly from the specified source IP address.

When non-owner access Telnet is disabled on a virtual router instance, Telnet sessions destined for the non-owner virtual router instance IP addresses are silently discarded in both master and backup modes.

### 3.4.3 Non-owner access SSH

When non-owner access SSH is enabled on a virtual router instance, authorized SSH sessions may be established that are destined for the virtual router instance IP addresses when operating in master mode. SSH sessions are always discarded at the IP interface when destined for a virtual router IP address operating in backup mode. Enabling non-owner access SSH does not guarantee SSH access; correct management and security features must be enabled to allow SSH on this interface and possibly from the specified source IP address. SSH is applicable to IPv4 VRRP only.

When non-owner access SSH is disabled on a virtual router instance, SSH sessions destined for the non-owner virtual router instance IP addresses are silently discarded in both master and backup modes.

## 3.5 VRRP instance inheritance

VRRP Instance Inheritance allows multiple VRRP instances to follow the state of a lead VRRP instance. This allows the VRRP control plane to handle more VRRP instances without requiring increased control message volumes.

The lead VRRP instance is configured by including the oper-group configuration statement within the lead VRRP instance configuration. The lead instance must be configured with the necessary message timers to detect VRRP failures at the configured rate. The following VRRP instances, referred to as following instances, are then associated with the appropriate lead VRRP instance by including the monitor-oper-group statement (for example, **monitor-oper-group “vrrp-LI-1”**).

The following are VRRP instance inheritance behaviors:

- One VRRP instance acts as the leading instance and behaves normally. This instance is configured with timers to attain the required detection times.
- The user can associate additional VRRP instances with the leading VRRP instance by configuring the following instances to monitor the lead oper-group instance.
- Command options associated with the instance state or priority are ignored within a following VRRP instance.
- If the lead instance becomes primary, all following instances assume a primary role for their respective VRRP instances.
- If the lead instance transitions from primary to standby, all the following instances transition to standby.
- If the lead instance transitions to a down state, all following instances transition to standby.

### 3.5.1 Configuration guidelines

The following guidelines apply to VRRP instance inheritance when configuring multiple VRRP instances that are bound together and share a common state with a lead VRRP instance.

- Nokia recommends all following VRRP instances exist on a common set of ports or LAG interface as the lead VRRP instance.
- Only include a single VRRP instance on the oper-group used for the lead VRRP instance.
- A VRRP instance cannot include both an oper-group and a monitor-oper-group simultaneously.
- A VRRP instance cannot be configured to monitor an oper-group and also be configured as passive.

## 3.5.2 VRRP instance inheritance configuration tasks

### 3.5.2.1 Lead VRRP instance configuration

Configure the lead VRRP instance with timer intervals for the wanted detection time. The key addition is the inclusion of the **oper-group** command to the configuration. The following example shows the lead VRRP instance configuration.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
    interface "base-1-1" {
        port 1/1/3:1
        ipv6 {
            link-local-address {
                address fe80::1
                duplicate-address-detection false
            }
            address 2500::1 {
                prefix-length 64
                duplicate-address-detection false
            }
            vrrp 1 {
                backup [2500::10 fe80::1:1]
                message-interval 5
                mac 00:00:5e:00:02:01
                priority 130
                ping-reply true
                oper-group "op-v6LI-1"
                bfd-liveness {
                    dest-ip 2000::2
                    service-name "100"
                    interface-name "bfd-1-1"
                }
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>router# info
#--------------------------------------------------
echo "IP Configuration"
#--------------------------------------------------
        interface "base-1-1"
            port 1/1/3:1
            ipv6
                address 2500::1/64 dad-disable
```

```
                link-local-address fe80::1 dad-disable
                vrrp 1
                    backup 2500::10
                    backup fe80::1:1
                    priority 130
                    ping-reply
                    message-interval 5
                    mac 00:00:5e:00:02:01
                    oper-group "op-v6LI-1"
                    bfd-enable name "100" interface "bfd-1-1" dst-ip 2000::2
                exit
            exit
            no shutdown
        exit
-----------------------------------------
```

### 3.5.2.2 Following VRRP instances

To configure VRRP instances with slower timer intervals include the **monitor-oper-group** command for MD-CLI and the **oper-group** command for classic CLI in the configuration. The following example shows VRRP instance configuration with slower timer intervals.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# interface base-1-2
     interface "base-1-2" {
        port 1/1/3:2
        ipv6 {
            link-local-address {
                address fe80::2
                duplicate-address-detection false
            }
            address 2500:0:1::1 {
                prefix-length 64
                duplicate-address-detection false
            }
            vrrp 1 {
                backup [2500:0:1::10 fe80::1:2]
                message-interval 40
                mac 00:00:5e:00:02:01
                priority 130
                ping-reply true
                monitor-oper-group "op-v6LI-1"
                bfd-liveness {
                    dest-ip 2000::2
                    service-name "100"
                    interface-name "bfd-1-1"
                }
            }
        }
    }
```

**Example: classic CLI**

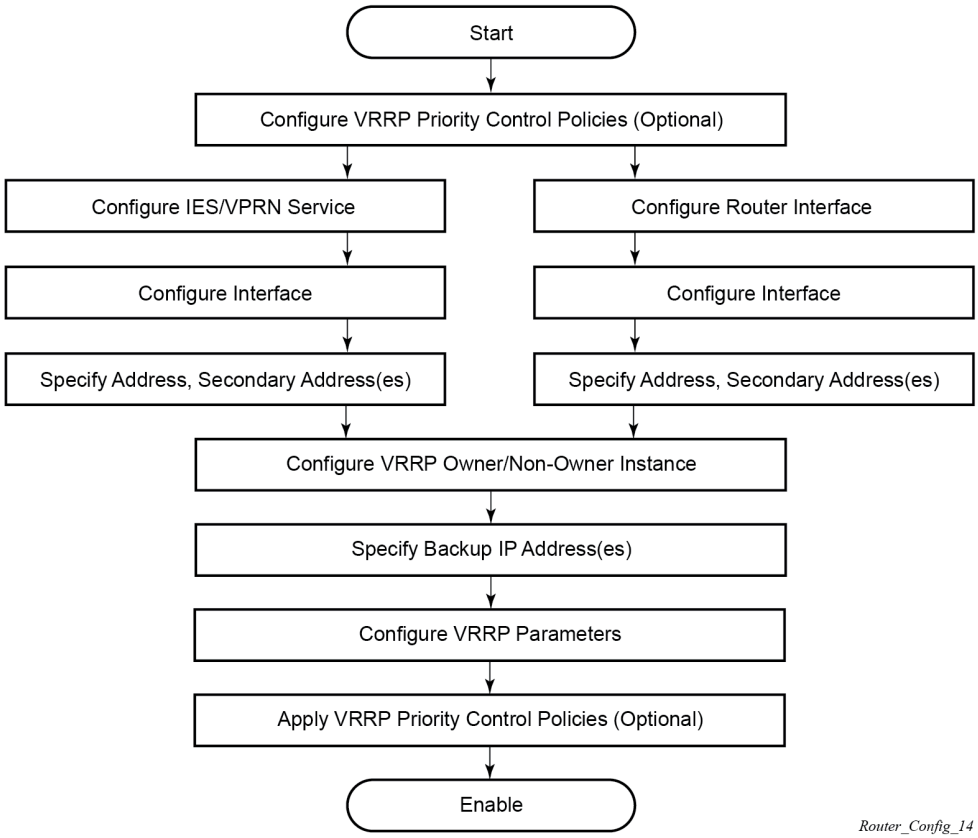
```
A:node-2>config>router# info
#--------------------------------------------------
echo "IP Configuration"
#--------------------------------------------------
        interface "base-1-2"
```

```
            port 1/1/3:2
            ipv6
                address 2500:0:1::1 dad-disable
                link-local-address fe80::1 dad-disable
                vrrp 1
                    backup 2500:0:1::10
                    backup fe80::1:2
                    message-interval 40
                    priority 130
                    ping-reply
                    mac 00:00:5e:00:02:01
                    oper-group "op-v6LI-1"
                    bfd-enable name "100" interface "bfd-1-1" dst-ip 2000::2
                exit
            exit
            no shutdown
        exit
```

## 3.6 VRRP configuration process overview

Figure 21: VRRP configuration and implementation flow shows the process to configure and implement VRRP command options.

*Figure 21: VRRP configuration and implementation flow*



## 3.7 Configuration notes

This section describes VRRP configuration restrictions.

### 3.7.1 General

- Creating and applying VRRP policies are optional.
- Backup command:
  - The backup IP addresses must be on the same subnet. The backup addresses explicitly define which IP addresses are in the VRRP advertisement message IP address list.
  - In the owner mode, the backup IP address must be identical to one of the interface's IP addresses. The backup address explicitly defines which IP addresses are in the VRRP advertisement message IP address list.
  - For IPv6, one of the backup addresses configured must be the link-local address of the owner VRRP instance.

## 3.8 Configuring VRRP with CLI

This section provides information to configure VRRP using the command line interface.

### 3.8.1 VRRP configuration overview

Configuring VRRP policies and configuring VRRP instances on interfaces and router interfaces is optional. The basic owner and non-owner VRRP configurations on an IES or router interface must specify the **backup** command option.

VRRP helps eliminate the single point of failure in a routed environment by using a virtual router IP address shared between two or more routers connecting the common domain. VRRP provides dynamic failover of the forwarding responsibility if the master becomes unavailable.

The VRRP implementation allows one master per IP subnet. All other VRRP instances in the same domain must be in backup mode.

#### 3.8.1.1 Preconfiguration requirements

Configuring VRRP policies:

VRRP policies must be configured before they can be applied to an interface or IES/VPRN VRRP instance. VRRP policies are configured in the **configure vrrp** context.

Configuring VRRP on an IES or VPRN service interface:

- The service customer account must be created before configuring an IES or VPRN VRRP instance.
- The interface address must be specified in the both the owner and non-owner IES, VPRN, or router interface instances.

## 3.8.2 Basic VRRP configurations

Configure VRRP command options in the following contexts.

### 3.8.2.1 VRRP policy

Configuring and applying VRRP policies are optional. There are no default VRRP policies. Each policy must be explicitly defined. A VRRP configuration must include the following:

- policy ID
- at least one of the following priority events:
  - port down
  - LAG port down
  - host unreachable
  - route unknown

**Example: VRRP policy configuration for the 7450 ESS (MD-CLI)**

```
[ex:/configure vrrp policy 100]
A:admin@node-2# info
    delta-in-use-limit 50
    priority-event {
        host-unreachable "10.10.24.4" {
            drop-count 25
        }
        lag-port-down "lag-1" {
            number-down 3 {
                priority {
                    priority-level 50
                    event-type explicit
                }
            }
        }
        port-down 4/1/2 {
            hold-set 43200
            priority {
                priority-level 100
                event-type delta
            }
        }
        port-down 4/1/3 {
            priority {
                priority-level 200
                event-type explicit
            }
        }
        route-unknown 10.10.0.0/32 {
            priority {
                priority-level 50
                event-type delta
                }
        }
    }
```

### Example: VRRP policy configuration for the 7450 ESS (classic CLI)

```
A:node-2>config>vrrp>policy# info
----------------------------------------------
            delta-in-use-limit 50
            priority-event
                port-down 4/1/2
                    hold-set 43200
                    priority 100 delta
                exit
                port-down 4/1/3
                    priority 200 explicit
                exit
                lag-port-down 1
                    number-down 3
                        priority 50 explicit
                    exit
                exit
                host-unreachable 10.10.24.4
                    drop-count 25
                exit
                route-unknown 10.10.0.0/32
                    priority 50 delta
                    exit
            exit
----------------------------------------------
```

### Example: VRRP policy configuration for the 7750 SR and 7950 XRS (MD-CLI)

```
[ex:/configure vrrp policy 100]
A:admin@node-2# info
    delta-in-use-limit 50
    priority-event {
        host-unreachable "10.10.24.4" {
            drop-count 25
        }
        port-down 4/1/2 {
            hold-set 43200
            priority {
                priority-level 100
                event-type delta
            }
        }
        port-down 4/1/3 {
            priority {
                priority-level 200
                event-type explicit
            }
        }
        route-unknown 10.10.0.0/32 {
            protocol [bgp]
            priority {
                priority-level 50
                event-type delta
            }
        }
    }
```

### Example: VRRP policy configuration for the 7750 SR and 7950 XRS (classic CLI)

```
A:node-2>config>vrrp>policy# info
----------------------------------------------
```

```
            delta-in-use-limit 50
            priority-event
                port-down 4/1/2
                    hold-set 43200
                    priority 100 delta
                exit
                port-down 4/1/3
                    priority 200 explicit
                exit
                lag-port-down 1
                    number-down 3
                        priority 50 explicit
                    exit
                exit
                host-unreachable 10.10.24.4
                    drop-count 25
                exit
                route-unknown 10.10.0.0/32
                    priority 50 delta
                    protocol bgp
                exit
            exit
-----------------------------------------------
```

### 3.8.2.2 VRRP IES service configuration

VRRP is configured within an IES service with two contexts: owner or nonowner. The user specifies the status when creating the VRRP configuration. When configured as owner, the virtual router instance (VRID) owns the backup IP addresses. All other virtual router instances participating in this message domain must have the same VRID configured and cannot be configured as owner.

SR OS supports passive VRRP, which does not require multiple VRRP instances to achieve default gateway load-balancing. Passive VRRP is a VRRP setting in which the transmission and reception of keepalive messages is completely suppressed; and therefore, the VPRN interface always behaves as the active router. Passive VRRP is enabled by adding the **passive** keyword to the VRRP instance at creation. For passive VRRP, the convergence time for link or node failures is not affected by the VRRP convergence, as all nodes in the VRRP instance are acting as active routers.

For IPv4, the user can configure up to four VRIDs on an IES service interface, with each of the four VRIDs able to manage up to 16 backup IP addresses. For IPv6, the user can configure four VRIDs on an IES service interface for FP4 cards and later, with each VRID able to manage up to 10 backup IP addresses.

VRRP is configured within an IES service and must include the following:

- VRID
- backup IP addresses

#### Example: IES service owner and nonowner VRRP configuration (MD-CLI)

```
[ex:/configure service ies "1"]
A:ex@node-2# info
    customer "1"
    interface "testing" {
        sap 1/1/55:0 {
        }
        ipv4 {
            primary {
                address 10.10.10.16
                prefix-length 24
```

```
            }
            vrrp 12 {
                authentication-key "z1rddawlDRZWzirCADyRv4MfzJVlDQsv hash2"
                backup [10.10.10.15]
                policy 1
            }
        }
    }
    interface "tuesday" {
        sap 7/1/1.2.2 {
        }
        ipv4 {
            primary {
                    address 10.10.36.2
                    prefix-length 24
                }
                vrrp 19 {
                    authentication-key "ungWv48Bz+pBQUDeXa4iI5Jsnw== hash2"
                    backup [10.10.36.2]
                    owner true
                }
            }
        }
    }
```

**Example: IES service owner and nonowner VRRP configuration (classic CLI)**

```
A:node-2>config>service# info
----------------------------------------------
...
        ies 1 name "1" customer 1 create
            interface "tuesday" create
                address 10.10.36.2/24
                sap 7/1/1.2.2 create
                vrrp 19 owner
                    backup 10.10.36.2
                    authentication-key "z1rddawlDRZWzirCADyRv4MfzJVlDQsv" hash2
                exit
            exit
            interface "testing" create
                address 10.10.10.16/24
                sap 1/1/55:0 create
                vrrp 12
                    backup 10.10.10.15
                    policy 1
                    authentication-key "ungWv48Bz+pBQUDeXa4iI5Jsnw==" hash2
                exit
            exit
            no shutdown
----------------------------------------------
```

### 3.8.2.2.1 Configure VRRP for IPv6

The following example shows a VRRP for IPV6 configuration and applies to the 7750 SR and 7950 XRS. The interface must be configured first.

**Example: MD-CLI**

```
[ex:/configure router "Base" ipv6 router-advertisement]
A:admin@node-2# info
```

```
    interface "Application-interface-101" {
        use-virtual-mac true
    }

[ex:/configure service ies "100"]
A:admin@node-2# info
    description "Application VLAN 921"
    customer "1"
    interface "Application-interface-101" {
        sap ccag-1.a:921 {
            description "cross connect to VPLS 921"
        }
        ipv4 {
            primary {
                address 10.152.2.220
                prefix-length 28
            }
            vrrp 217 {
                backup [10.152.2.222]
                priority 254
                ping-reply true
            }
        }
        ipv6 {
            link-local-address {
                address fe80::d68f:1:221:fffd
                duplicate-address-detection false
            }
            address 2001:db8:d68f:1:221::fffd {
                prefix-length 64
            }
            vrrp 219 {
                backup [fe80::d68f:1:221:ffff]
                priority 254
                ping-reply true
            }
        }
    }
```

### Example: classic CLI

```
A:node-2>config>router>router-advert# info
----------------------------------------------
            interface "Application-interface-101"
                use-virtual-mac
                no shutdown
            exit
...
----------------------------------------------

A:node-2>config>service>ies# info
----------------------------------------------
            description "Application VLAN 921"
            interface "Application-interface-101" create
                address 10.152.2.220/28
                vrrp 217
                    backup 10.152.2.222
                    priority 254
                    ping-reply
                exit
                ipv6
                    address 2001:db8:D68F:1:221::FFFD/64
                    link-local-address fe80::d68f:1:221:fffd dad-disable
```

```
                    vrrp 219
                        backup fe80::d68f:1:221:ffff
                        priority 254
                        ping-reply
                    exit
                exit
                sap ccag-1.a:921 create
                    description "cross connect to VPLS 921"
                exit
            exit
            no shutdown
----------------------------------------------
```

### 3.8.2.3 VRRP router interface command options

VRRP command options are configured on a router interface with two contexts: owner or nonowner. The user specifies the status is specified when creating the VRRP configuration. When configured as owner, the virtual router instance (VRID) owns the backed up IP addresses. All other virtual router instances participating in this message domain must have the same VRID configured and cannot be configured as owner.

SR OS supports passive VRRP, which does not require multiple VRRP instances to achieve default gateway load-balancing. Passive VRRP is a VRRP setting in which the transmission and reception of keepalive messages is completely suppressed; and therefore, the VPRN interface always behaves as the active router. Passive VRRP is enabled by adding the **passive** keyword to the VRRP instance at creation. For passive VRRP, the convergence time for link or node failures is not affected by the VRRP convergence, as all nodes in the VRRP instance are acting as active routers.

For IPv4, the user can configure up to four VRIDs on a router interface, with each VRID able to manage up to 16 backup IP addresses. For IPv6, the user can configure four VRIDs on a router interface for FP4 cards and later, with each VRID able to manage up to 10 backup IP addresses.

VRRP command options configured on a router interface must include the following:

- VRID
- backup IP addresses

#### Example: Router interface owner and nonowner VRRP configuration (MD-CLI)

```
[ex:/configure router "Base"]
A:admin@node-2#
...
    interface "system" {
    ipv4 {
            primary {
                address 10.10.0.4 {
                prefix-length 32
            }
        }
    }
    interface "test1" {
        ipv4 {
            primary {
                address 10.10.14.1
                prefix-length 24
            }
            secondary 10.10.16.1 {
                prefix-length 24
            }
```

```
            secondary 10.10.17.1 {
                prefix-length 24
            }
            secondary 10.10.18.1 {
                prefix-length 24
            }
        }
    }
    interface "test2" {
        ipv4 {
            primary {
                address 10.10.10.23
                prefix-length 24
            }
            vrrp 1 {
                authentication-key "V+mMCSI1pnX+5dHDE729xj4E3YCngRQ= hash2"
                backup [10.10.10.23]
                owner true
            }
        }
    }
```

**Example: Router interface owner and nonowner VRRP configuration (classic CLI)**

```
A:node-2>config>router# info
#------------------------------------------
echo "IP Configuration "
#------------------------------------------
        interface "system"
            address 10.10.0.4/32
        exit
        interface "test1"
            address 10.10.14.1/24
            secondary 10.10.16.1/24
            secondary 10.10.17.1/24
            secondary 10.10.18.1/24
        exit
        interface "test2"
            address 10.10.10.23/24
            vrrp 1 owner
                backup 10.10.10.23
                authentication-key "V+mMCSI1pnX+5dHDE729xj4E3YCngRQ=" hash2
            exit
        exit
#------------------------------------------
```

### 3.8.3 Common configuration tasks

This section provides a brief overview of the tasks that must be performed to configure VRRP and provides the CLI commands.

VRRP command options are defined under a service interface or a router interface context. An IP address must be assigned to each IP interface. Only one IP address can be associated with an IP interface but several secondary IP addresses also be associated.

Owner and non-owner configurations must include the following command options:

- All participating routers in a VRRP instance must be configured with the same VRID.

- All participating non-owner routers can specify up to 16 backup IP addresses (IP addresses that the master is representing). The owner configuration must include at least one backup IP address.
- For IPv6, all participating routers must be configured with the same link-local backup address (the one configured for the owner instance).

Other owner and non-owner configurations include the following optional commands:

- authentication-key
- MAC
- message-interval

In addition to the common command options, the following non-owner commands can be configured:

- master-int-inherit
- ntp-reply
- priority
- policy
- ping-reply
- preempt
- telnet-reply
- ssh-reply (IPv4 only)
- [no] shutdown

### 3.8.3.1 Creating interface command options

If multiple subnets are configured on an Ethernet interface, VRRP can be configured on each subnet.

The following example shows an IP interface configuration.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
    router-id 10.10.0.1
    interface "system" {
    ipv4 {
            primary {
                address 10.10.0.1 {
                prefix-length 32
            }
        }
    }
    interface "testA" {
        ipv4 {
            primary {
                address 10.123.123.123
                prefix-length 24
            }
        }
    }
     interface "testB" {
        ipv4 {
            primary {
                address 10.10.14.1
```

```
                prefix-length 24
            }
            secondary 10.10.16.1 {
                prefix-length 24
            }
            secondary 10.10.17.1 {
                prefix-length 24
            }
            secondary 10.10.18.1 {
                prefix-length 24
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>router# info
#------------------------------------------
echo "IP Configuration "
#------------------------------------------
        interface "system"
            address 10.10.0.1/32
        exit
        interface "testA"
            address 10.123.123.123/24
        exit
        interface "testB"
            address 10.10.14.1/24
            secondary 10.10.16.1/24
            secondary 10.10.17.1/24
            secondary 10.10.18.1/24
        exit
        router-id 10.10.0.1
#------------------------------------------
```

### 3.8.4 Configuring VRRP policy components

The following example shows VRRP policy component configurations.

**Example: MD-CLI**

```
[ex:/configure vrrp policy 1]
A:admin@node-2# info
    delta-in-use-limit 50
    priority-event {
        port-down 1/1/2 {
            hold-set 43200
            priority {
                priority-level 100
                event-type delta
            }
        }
        route-unknown 0.0.0.0/0 {
            protocol [isis]
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>vrrp# info
----------------------------------------------
        policy 1
            delta-in-use-limit 50
            priority-event
                port-down 1/1/2
                    hold-set 43200
                    priority 100 delta
                exit
                route-unknown 0.0.0.0/0
                    protocol isis
                exit
            exit
        exit
----------------------------------------------
```

### 3.8.4.1 Configuring service VRRP

VRRP command options can be configured on an interface in a service to provide virtual default router support, which allows traffic to be routed without relying on a single router in case of failure. VRRP can be configured in the following two ways.

#### 3.8.4.1.1 Non-owner VRRP

The following example shows a basic non-owner VRRP configuration.

**Example: MD-CLI**

```
[ex:/configure service ies "100"]
A:admin@node-2# info
    interface "testing" {
        sap 1/1/55:0 {
         ipv4 {
            primary {
                address 10.10.10.16
                prefix-length 24
            }
        ipv4 {
            vrrp 12 {
                authentication-key "testabc"
                backup [10.10.10.15]
                policy 1
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>service>ies# info
----------------------------------------------
            interface "testing" create
                address 10.10.10.16/24
                sap 1/1/55:0 create
                vrrp 12
                    backup 10.10.10.15
```

```
                    policy 1
                    authentication-key "testabc"
                exit
            exit
-----------------------------------------
```

#### 3.8.4.1.2 Owner service VRRP

The following example shows an owner service VRRP configuration.

**Example: MD-CLI**

```
[ex:/configure router "Base" interface "test2"]
A:admin@node-2# info
    ipv4 {
        primary {
            address 10.10.10.23
            prefix-length 24
        }
        vrrp 1 {
            authentication-key "testabc"
            backup [10.10.10.23]
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>router# info
#------------------------------------------
echo "IP Configuration "
#------------------------------------------
...
        interface "test2"
            address 10.10.10.23/24
            vrrp 1 owner
                backup 10.10.10.23
                authentication-key "testabc"
            exit
        exit
#------------------------------------------
```

### 3.8.4.2 Configuring router interface VRRP command options

VRRP command options can be configured on an interface in an interface to provide virtual default router support, which allows traffic to be routed without relying on a single router in case of failure. VRRP can be configured in following two ways.

#### 3.8.4.2.1 Router interface VRRP non-owner

The following example shows a router interface VRRP non-owner configuration.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2# info
```

```
        interface "if-test" {
            ipv4 {
                primary {
                    address 10.20.30.40
                    prefix-length 24
                }
                secondary 10.10.50.1 {
                    prefix-length 24
                }
                secondary 10.10.60.1 {
                    prefix-length 24
                }
                secondary 10.10.70.1 {
                    prefix-length 24
                }
                vrrp 1 {
                    authentication-key "testabc hash2"
                    backup [10.10.50.2 10.10.60.2 10.10.70.2 10.20.30.41]
                    ping-reply true
                    telnet-reply true
                }
            }
```

**Example: classic CLI**

```
A:node-2>config>router# info
#------------------------------------------
     interface "if-test"
            address 10.20.30.40/24
            secondary 10.10.50.1/24
            secondary 10.10.60.1/24
            secondary 10.10.70.1/24
            vrrp 1
                backup 10.10.50.2
                backup 10.10.60.2
                backup 10.10.70.2
                backup 10.20.30.41
                ping-reply
                telnet-reply
                authentication-key "testabc" hash2
            exit
        exit
#------------------------------------------
```

### 3.8.4.2.2 Router interface VRRP owner

The following example shows a router interface VRRP owner configuration.

**Example: MD-CLI**

```
[ex:/configure router "Base"]
A:admin@node-2#
    }
    interface "vrrpowner" {
        ipv4 {
            primary {
                address 10.10.10.23
                prefix-length 24
            }
            vrrp 1 {
```

```
                authentication-key "testabc hash2"
                backup [10.10.10.23]
                owner true
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>router# info
#------------------------------------------
     interface "vrrpowner"
            address 10.10.10.23/24
            vrrp 1 owner
                backup 10.10.10.23
                authentication-key "testabc" hash2
            exit
            no shutdown
        exit
#------------------------------------------
```

## 3.9 VRRP configuration management tasks

This section describes VRRP configuration management tasks.

### 3.9.1 Modifying a VRRP policy

To access a specific VRRP policy, specify the policy ID. Use the following command to display a list of VRRP policies.

```
show vrrp policy
```

The following example shows the modified VRRP policy configuration.

**Example: MD-CLI**

```
[ex:/configure vrrp policy 100]
A:admin@node-2# info
    delta-in-use-limit 50
    priority-event {
        host-unreachable "10.10.24.4" {
            drop-count 25
        }
        port-down 1/1/2 {
            hold-set 43200
        }
        port-down 1/1/3 {
            priority {
                priority-level 200
                event-type explicit
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>vrrp>policy# info
----------------------------------------------
            delta-in-use-limit 50
            priority-event
                port-down 1/1/2
                    hold-set 43200
                    priority 100 delta
                exit
                port-down 1/1/3
                    priority 200 explicit
                exit
                host-unreachable 10.10.24.4
                    drop-count 25
                exit
            exit
----------------------------------------------
```

### 3.9.1.1 Deleting a VRRP policy

VRRP policies are only applied to non-owner VRRP instances. A VRRP policy cannot be deleted if it is applied to an interface or to an IES service. Each instance in which the policy is applied must be deleted.

Use the following command to show where VRRP policies are applies to an entity.

```
show vrrp policy
```

The following example shows where VRRP policies are applied to an entity.

**Output example: MD-CLI**

```
===============================================================================
VRRP Policies
===============================================================================
Policy    Current             Current      Current      Delta Applied Svc
Id        Priority & Effect   Explicit     Delta Sum    Limit         Context
-------------------------------------------------------------------------------
1         200 Explicit        200          100          50          Yes
15        254                 None         None         1           No
32        100                 None         None         1           No
===============================================================================
```

**Output example: classic CLI**

```
===============================================================================
VRRP Policies
===============================================================================
Policy    Current             Current      Current      Delta       Applied
Id        Priority & Effect   Explicit     Delta Sum    Limit
-------------------------------------------------------------------------------
1         200 Explicit        200          100          50          Yes
15        254                 None         None         1           No
32        100                 None         None         1           No
===============================================================================
```

The following example shows the deletion of a VRRP policy.

**Example: MD-CLI**

```
*[ex:/configure vrrp]
A:admin@node-2# delete policy 1

*[ex:/configure vrrp]
A:admin@node-2# commit
```

**Example: classic CLI**

```
*A:node-2>config>vrrp# no policy 1
```

## 3.9.2 Modifying service and interface VRRP command options

### 3.9.2.1 Modifying non-owner command options

After a VRRP instance is created as non-owner, it cannot be modified to the owner state. The VRID must be deleted, then recreated with the **owner** keyword, to invoke IP address ownership.

### 3.9.2.2 Modifying owner command options

After a VRRP instance is created as **owner**, it cannot be modified to the non-owner state. The VRID must be deleted, then recreated without the owner keyword, to remove IP address ownership.

Entering the **owner** command option is optional when entering the VRID for modification purposes.

### 3.9.2.3 Deleting VRRP from an interface or service

The VRID does not need to be shutdown to remove the virtual router instance from an interface or service. Use the following command to remove the virtual router instance:

- **MD-CLI**

```
configure service ies interface ipv4 delete vrrp
```

- **classic CLI**

```
configure service ies interface vrrp shutdown
configure service ies interface no vrrp
```

# 4 Filter policies

## 4.1 ACL filter policy overview

ACL filter policies, also referred to as Access Control Lists (ACLs) or just "filters", are sets of ordered rule entries specifying packet match criteria and actions to be performed to a packet upon a match. Filter policies are created with a unique filter ID and filter name. The filter name needs to be assigned during the creation of the filter policy. If a name is not specified at creation time, then SR OS assigns a string version of the filter ID as the name.

There are three main filter policies: **ip-filter** for IPv4, **ipv6-filter** for IPv6, and **mac-filter** for MAC level filtering. Additionally, the filter policy **scope** defines if the policy can be reused between different interfaces, embedded in another filter policy or applied at the system level:

- **exclusive filter**

  An exclusive filter defines policy rules explicitly for a single interface. An exclusive filter allows the highest level of customization but uses the most resources, because each exclusive filter consumes hardware resources online cards on which the interface exists.

- **template filter**

  A template filter uses an identical set of policy rules across multiple interfaces. Template filters use a single set of resources per line card, regardless of how many interfaces use a specific template filter policy on that line card. Template filter policies used on access interfaces consume resources online cards only if at least one access interface for a specific template filter policy is configured on a specific line card.

- **embedded filter**

  An embedded filter defines a common set of policy rules that can then be used (embedded) by other exclusive or template filters in the system. This allows optimized management of filter policies.

- **system filter**

  A system filter policy defines a common set of policy rules that can then be activated within other exclusive/template filters. It can be used, for example, as a system-level set of deny rules. This allows optimized management of common rules (similarly to embedded filters). However, active system filter policy entries are not duplicated inside each policy that activates the system policy (as is the case when embedding is used). The active system policy is downloaded after to line cards, and activated filter policies are chained to it.

After the filter policy is created, the policy must then be associated with interfaces, services, subscribers, or with other filter policies (if the created policy cannot be directly deployed on an interface, service, or subscriber), so the incoming or outgoing traffic can be subjected to filter rules. Filter policies are associated with interfaces, services, or subscribers separately in the ingress and egress directions. A policy deployed on ingress and egress direction can be the same or different. In general, Nokia recommends using different filter policies for the ingress and egress directions and to use different filter policies per service type, because filter policies support different match criteria and different actions for different directions/service contexts.

A filter policy is applied to a packet in the ascending rule entry order. When a packet matches all the command options specified in a filter entry's match criteria, the system takes the action defined for that entry. If a packet does not match the entry command options, the packet is compared to the next higher numerical filter entry rule, and so on.

In classic CLI, if the packet does not match any of the entries, the system executes the default action specified in the filter policy: drop or forward.

In MD-CLI, if the packet does not match any of the entries, the system executes the default action specified in the filter policy: drop or accept.

For Layer 2, either an IPv4/IPv6 or MAC filter policy can be applied. For Layer 3 and network interfaces, an IPv4/IPv6 policy can be applied. For R-VPLS service, a Layer 2 filter policy can be applied to Layer 2 forwarded traffic and a Layer 3 filter policy can be applied to Layer 3 routed traffic. For dual-stack interfaces, if both IPv4 and IPv6 filter policies are configured, the policy applied are based on the outer IP header of the packet. Non-IP packets do not affect an IP filter policy, so the default action in the IP filter policy do not apply to these packets. Egress IPv4 QoS-based classification criteria are ignored when egress MAC filter policy is configured on the same interface.

Additionally, platforms that support Network Group Encryption (NGE) can use IP exception filters. IP exception filters scan all outbound traffic entering an NGE domain and allow packets that match the exception filter criteria to transit the NGE domain unencrypted. See Router encryption exceptions using ACLs for information about IP exception filters supported by NGE nodes.

### 4.1.1 Filter policy basics

The following subsections define main functionality supported by filter policies.

#### 4.1.1.1 Filter policy packet match criteria

This section defines packet match criteria supported on SR OS for IPv4, IPv6, and MAC filters. Supported criteria types depend on the hardware platform and filter direction, see your Nokia representative for more information.

General notes:

- If multiple unique match criteria are specified in a single filter policy entry, all criteria must be met in order for the packet to be considered a match against that filter policy entry (logical AND).
- Any match criteria not explicitly defined is ignored during match.
- An ACL filter policy entry with match criteria defined, but no action configured, is considered incomplete and inactive (an entry is not downloaded to the line card). A filter policy must have at least one entry active for the policy to be considered active.
- An ACL filter entry with no match conditions defined matches all packets.
- Because an ACL filter policy is an ordered list, entries should be configured (numbered) from the most explicit to the least explicit.

#### 4.1.1.2 IPv4/IPv6 filter policy entry match criteria

This section describes the IPv4 and IPv6 match criteria supported by SR OS. The criteria are evaluated against the outer IPv4 or IPv6 header and a Layer 4 header that follows (if applicable). Support for match

criteria may depend on hardware or filter direction. Nokia recommends not configuring a filter in a direction or on hardware where a match criterion is not supported because this may lead to unwanted behavior.

IPv4 and IPv6 filter policies support three four filter types, including normal, source MAC, packet length, and destination class, with each supporting a different set of match criteria.

The match criteria available using the normal filter type are defined in this section. Layer 3 match criteria include:

- **DSCP**

  Match the specified DSCP command option against the Differentiated Services Code Point/Traffic Class field in the IPv4 or IPv6 packet header.

- **source IP, destination IP, or IP**

  Match the specified source or destination IPv4 or IPv6 address prefix against the IP address field in the IPv4 or IPv6 packet header. The user can optionally configure a mask to be used in a match. The **ip** command can be used to configure a single filter-policy entry that provides non-directional matching of either the source or destination (logical OR).

- **flow label**

  Match the specified flow label against the Flow label field in IPv6 packets. The user can optionally configure a mask to be used in a match. This operation is supported on ingress filters.

- **protocol**

  Match the specified protocol against the Protocol field in the IPv4 packet header (for example, TCP, UDP, IGMP) of the outer IPv4. "*" can be used to specify TCP or UDP upper-layer protocol match (Logical OR).

- **Next Header**

  Match the specified upper-layer protocol (such as, TCP, UDP, IGMPv6) against the Next Header field of the IPv6 packet header. "*" can be used to specify TCP or UDP upper-layer protocol match (Logical OR).

  Use the following command to match against up to six extension headers.

  ```
  configure system ip ipv6-eh max
  ```

  Use the following command to match against the Next Header value of the IPv6 header.

  ```
  configure system ip ipv6-eh limited
  ```

### Fragment match criteria

Match for the presence of fragmented packet. For IPv4, match against the MF bit or Fragment Offset field to determine whether the packet is a fragment. For IPv6, match against the Next Header field for the Fragment Extension Header value to determine whether the packet is a fragment. Up to six extension headers are matched against to find the Fragmentation Extension Header.

IPv4 and IPv6 filters support matching against initial fragment using **first-only** or non-initial fragment **non-first-only**.

IPv4 match fragment **true** or **false** criteria are supported on both ingress and egress.

IPv4 match fragment **first-only** or **non-first-only** are supported on ingress only.

### Operational note for fragmented traffic

IP and IPv6 filters defined to match TCP, UDP, ICMP, or SCTP criteria (such as source port, destination port, port, TCP ACK, TCP SYN, ICMP type, ICMP code) with command options of zero or false also match non-first fragment packets if other match criteria within the same filer entry are also met. Non-initial fragment packets do not contain a UDP, TCP, ICMP or SCTP header.

### IPv4 options match criteria

You can configure the following IPv4 options match criteria exist:

- **IP option**

  Matches the specified command option value in the first option of the IPv4 packet. A user can optionally configure a mask to be used in a match.

- **option present**

  Matches the presence of IP options in the IPv4 packet. Padding and EOOL are also considered as IP options. Up to six IP options are matched against.

- **multiple option**

  Matches the presence of multiple IP options in the IPv4 packet.

- **source route option**

  Matches the presence of IP Option 3 or 9 (Loose or Strict Source Route) in the first three IP options of the IPv4 packet. A packet also matches this rule if the packet has more than three IP options.

### IPv6 extension header match criteria

You can configure the following IPv6 Extension Header match criteria:

- **Authentication Header extension header**

  Matches for the presence of the Authentication Header extension header in the IPv6 packet. This match criterion is supported on ingress only.

- **Encapsulating Security Payload extension header**

  Matches for the presence of the Encapsulating Security Payload extension header in the IPv6 packet. This match criterion is supported on ingress only.

- **hop-by-hop options**

  Matches for the presence of hop-by-hop options extension header in the IPv6 packet. This match criterion is supported on ingress only.

- **Routing extension header type 0**

  Matches for the presence of Routing extension header type 0 in the IPv6 packet. This match criterion is supported on ingress only.

### Upper-layer protocol match criteria

You can configure the following upper-layer protocol match criteria:

- **ICMP/ICMPv6 code field header**

  Matches the specified value against the code field of the ICMP or ICMPv6 header of the packet. This match is supported only for entries that also define protocol or next-header match for the ICMP or ICMPv6 protocol.

- **ICMP/ICMPv6 type field header**

Matches the specified value against the type field of the ICMP or ICMPv6 header of the packet. This match is supported only for entries that also define the protocol or next-header match for the ICMP or ICMPv6 protocol.

- **source port number, destination port number, or port**

  Matches the specified port, port list, or port range against the source port number or destination port number of the UDP, TCP, or SCTP packet header. An option to match either source or destination (Logical OR) using a single filter policy entry is supported by using a directionless port. Source or destination match is supported only for entries that also define protocol/next-header match for TCP, UDP, SCTP, or TCP or UDP protocols. Match on SCTP source port, destination port, or port is supported on ingress filter policy.

- **TCP ACK, TCP CWR, TCP ECE, TCP FIN, TCP NS, TCP PSH, TCP RST, TCP SYN, TCP URG**

  Matches the presence or absence of the TCP flags defined in RFC 793, RFC 3168, and RFC 3540 in the TCP header of the packet. This match criteria also requires defining the protocol/next-header match as TCP in the filter entry. TCP CWR, TCP ECE, TCP FIN, TCP NS, TCP PSH, TCP URG are supported on FP4 and FP5-based line cards only. TCP ACK, TCP RST, and TCP SYN are supported on all FP-based cards. When configured on other line cards, the bit for the unsupported TCP flags is ignored.

- **tcp-established**

  Matches the presence of the TCP flags ACK or RST in the TCP header of the packet. This match criteria requires defining the protocol/next-header match as TCP in the filter entry.

**Note:** Non initial fragmented packets do not match filter entries configured with layer 4 header match criteria. Only the first fragment of a packet includes the layer 4 header information.

#### For filter type match criteria

Additional match criteria for source MAC, packet length, and destination class are available using different filter types. See Filter policy type for more information.

IP prefixes, protocol numbers, TCP-UDP ports, and packet length values or ranges can also be grouped in the command **configure filter match-list**; see Filter policy advanced topics for more information.

### 4.1.1.3 MAC filter policy entry match criteria

MAC filter policies support three different filter types with normal, ISID, and VID each supporting a different set of match criteria.

The following list describes the MAC match criteria supported by SR OS or switches for all types of MAC filters (normal, ISID, and VID). The criteria are evaluated against the Ethernet header of the Ethernet frame. Support for match criteria may depend on H/W or filter direction as described in the following description. Match criteria is blocked if it is not supported by a specified frame-type or MAC filter type. Nokia recommends not configuring a filter in a direction or on hardware where a match condition is not supported as this may lead to unwanted behavior.

You can configure the following MAC filter policy entry match criteria:

- **frame format**

  The filter searches to match a specific type of frame format. For example, configuring frame-type ethernet_II matches only ethernet-II frames.

- **source MAC address**

The filter searches to match source MAC address frames. The user can optionally configure a mask to be used in a match.

- **destination MAC address**

  The filter searches to match destination MAC address frames. The user can optionally configure a mask to be used in a match.

- **802.1p frames**

  The filter searches to match 802.1p frames. The user can optionally configure a mask to be used in a match.

- **Ethernet II frames**

  The filter searches to match Ethernet II frames. The Ethernet type field is a two-byte field used to identify the protocol carried by the Ethernet frame.

- **source access point**

  The filter searches to match frames with a source access point on the network node designated in the source field of the packet. The user can optionally configure a mask to be used in a match.

- **destination access point**

  The filter searches to match frames with a destination access point on the network node designated in the destination field of the packet. The user can optionally configure a mask to be used in a match.

- **specified three-byte OUI**

  The filter searches to match frames with the specified three-byte OUI field.

- **specified two-byte protocol ID**

  The filter searches to match frames with the specified two-byte protocol ID that follows the three-byte OUI field.

- **ISID**

  The filter searches to match for the matching Ethernet frames with the 24-bit ISID value from the PBB I-TAG. This match criterion is mutually exclusive of all the other match criteria under a specific MAC filter policy and is applicable to MAC filters of type ISID only. The resulting MAC filter can only be applied on a BVPLS SAP or PW in the egress direction.

- **inner-tag or outer-tag**

  The filter searches to match Ethernet frames with the non-service delimiting tags, as described in the VID MAC filters section. This match criterion is mutually exclusive of all other match criteria under a specific MAC filter policy and is applicable to MAC filters of type VID only.

#### 4.1.1.4 IP exception filters

An NGE node supports IPv4 exception filters. See Router encryption exceptions using ACLs for information about IP exception filters supported by NGE nodes.

#### 4.1.1.5 Filter policy actions

The following actions are supported by ACL filter policies:

- **drop**

Allows users to deny traffic to ingress or egress the system.

– **IPv4 packet-length and IPv6 payload-length conditional drop**

  Traffic can be dropped based on IPv4 packet length or IPv6 payload length by specifying a packet length or payload length value or range within the drop filter action (the IPv6 payload length field does not account for the size of the fixed IP header, which is 40 bytes).

  This filter action is supported on ingress IPv4 and IPv6 filter policies only, if the filter is configured on an egress interface the **packet-length** or **payload-length** match condition is always true.

  This **drop** condition is a filter entry action evaluation, and not a filter entry match evaluation. Within this evaluation, the condition is checked after the packet matches the entry based on the specified filter entry match criteria.

  Packets that match a filter policy entry match criteria and the **drop** *packet-length-value* or *payload-length-value* are dropped. Packets that match only the filter policy entry match criteria and do not match the **drop** *packet-length-value* or **drop** *payload-length value* are forwarded with no further matching in following filter entries.

  Packets matching this filter entry and not matching the conditional criteria are not logged, counted, or mirrored.

– **IPv4 TTL and IPv6 hop limit conditional drop**

  Traffic can be dropped based on a IPv4 TTL or IPv6 hop limit by specifying a TTL or hop limit value or range within the **drop** filter action.

  This filter action is supported on ingress IPv4 and IPv6 filter policies only. If the filter is configured on an egress interface the packet-length or payload-length match condition is always true.

  This **drop** condition is a filter entry action evaluation, and not a filter entry match evaluation. Within this evaluation, the condition is checked after the packet matches the entry based on the specified filter entry match criteria.

  Packets that match filter policy entry match criteria and the drop TTL or drop hop limit value are dropped. Packets that match only the filter policy entry match criteria and do not match the drop TTL value or drop hop limit value are forwarded with no further match in following filter entries.

  Packets matching this filter entry and not matching the conditional criteria are not logged, counted, or mirrored.

– **pattern conditional drop**

  Traffic can be dropped when it is based on a pattern found in the packet header or data payload. The pattern is defined by an expression, mask, offset type, and offset value match in the first 256 bytes of a packet.

  The pattern expression is up to 8 bytes long. The **offset-type** command identifies the starting point for the offset value and the supported offset-type command options are:

  - **layer-3**: layer 3 IP header
  - **layer-4**: layer 4 protocol header
  - **data**: data payload for TCP or UDP protocols
  - **dns-qtype**: DNS request or response query type

  The content of the packet is compared with the expression/mask value found at the offset type and offset value as defined in the filter entry. For example, if the pattern is expression 0xAA11, mask 0xFFFF, offset-type data, offset-value 20, the filter entry compares the content of the first 2 bytes in the packet data payload found 20 bytes after the TCP/UDP header with 0xAA11.

This drop condition is a filter entry action evaluation, and not a filter entry match evaluation. Within this evaluation, the condition is checked after the packet matches the entry based on the specified filter entry match criteria.

Packets that match a filter policy's entry match criteria and the pattern, are dropped. Packets that match only the filter policy's entry match criteria and do not match the pattern, are forwarded without a further match in subsequent filter entries.

This filtering capability is supported on ingress IPv4 and IPv6 policies using FP4-based line cards, and cannot be configured on egress. A filter entry using a pattern, is not supported on FP2 or FP3-based line cards. If programmed, the pattern is ignored and the action is forward.

Packets matching this filter entry and not matching the conditional criteria are not logged, counted, or mirrored.

- **drop extracted traffic**

  Traffic extracted to the CPM can be dropped using ingress IPv4 and IPv6 filter policies based on filter match criteria. Any IP traffic extracted to the CPM is subject to this filter action, including routing protocols, snooped traffic, and TTL expired traffic.

  Packets that match the filter entry match criteria and extracted to the CPM are dropped. Packets that match only the filter entry match criteria and are not extracted to the CPM are forwarded with no further match in the subsequent filter entries.

  Cflowd, log, mirror, and statistics apply to all traffic matching the filter entry, regardless of the drop or forward action.

- **forward**

  Allows users to accept traffic to ingress or egress the system and be subject to regular processing.

- **accept a conditional filter action**

  Allows users to accept a conditional filter action. Use the following commands to configure a conditional filter action:

  – **MD-CLI**

  ```
  configure filter ip-filter entry action accept-when
  configure filter ipv6-filter entry action accept-when
  ```

  – **classic CLI**

  ```
  configure filter ip-filter entry action forward-when
  configure filter ipv6-filter entry action forward-when
  ```

  – **pattern conditional accept**

  Traffic can be accepted based on a pattern found in the packet header or data payload. The pattern is defined by an expression, mask, offset type, and offset value match in the first 256 bytes of a packet. The pattern expression is up to 8 bytes long. The offset type identifies the starting point for the offset value and the supported offset types are:

  - **layer-3**: Layer 3 IP header
  - **layer-4**: Layer 4 protocol header
  - **data**: data payload for TCP or UDP protocols
  - **dns-qtype**: DNS request or response query type

– The content of the packet is compared with the expression/mask value found at the offset type and offset value defined in the filter entry. For example, if the pattern is expression 0xAA11, mask 0xFFFF, offset-type data, and offset-value 20, then the filter entry compares the content of the first 2 bytes in the packet data payload found 20 bytes after the TCP/UDP header with 0xAA11.

  This accept condition is a filter entry action evaluation, and not a filter entry match evaluation. Within this evaluation, the condition is checked after the packet matches the entry based on the specified filter entry match criteria. Packets that match a filter policy's entry match criteria and the pattern, are accepted. Packets that match only the filter policy's entry match criteria and do not match the pattern, are dropped without a further match in subsequent filter entries.

  This filtering capability is supported on ingress IPv4 and IPv6 policies using FP4-based line cards and cannot be configured on egress. A filter entry using a pattern is not supported on FP2 or FP3-based line cards. If programmed, the pattern is ignored and the action is drop.

  Packets matching this filter entry and not matching the conditional criteria are not logged, counted, or mirrored.

- **FC**

  Allows users to mark the forwarding class (FC) of packets. This command is supported on ingress IP and IPv6 filter policies. This filter action can be combined with the rate-limit action.

  Packets matching this filter entry action bypass QoS FC marking and are still subject to QoS queuing, policing and priority marking.

  The QPPB forwarding class takes precedence over the filter FC marking action.

- **rate limit**

  This action allows users to rate limit traffic matching a filter entry match criteria using IPv4, IPv6, or MAC filter policies.

  If multiple interfaces (including LAG interfaces) use the same **rate-limit** filter policy on different FPs, then the system allocates a rate limiter resource for each FP; an independent rate limit applies to each FP.

  If multiple interfaces (including LAG interfaces) use the same **rate-limit** filter policy on the same FP, then the system allocates a single rate limiter resource to the FP; a common aggregate rate limit is applied to those interfaces.

  Note that traffic extracted to the CPM is not rate limited by an ingress **rate-limit** filter policy while any traffic generated by the router can be rate limited by an egress **rate-limit** filter policy.

  **rate-limit** filter policy entries can coexist with cflowd, log, and mirror regardless of the outcome of the rate limit. This filter action is not supported on egress on 7750 SR-a.

  Rate limit policers are configured with the maximum burst size (MBS) equals the committed burst size (CBS) equals 10 ms of the rate and high-prio-only equals 0.

  Interaction with QoS: Packets matching an ingress **rate-limit** filter policy entry bypass ingress QoS queuing or policing, and only the filter rate limit policer is applied. Packets matching an egress **rate-limit** filter policy bypass egress QoS policing, normal egress QoS queuing still applies.

  – **Kilobits-per-second and packets-per-second rate limit**
    The rate-limit action can be defined using kilobits per second or packets per second and is supported on both ingress and egress filter policies. The MBS value can also be configured using the kilobits-per-second policer.

    The packets-per-second rate limit and kilobits-per-second MBS are not supported when using a MAC filter policy and not supported on 7750 SR-a.

– **IPv4 packet-length and IPv6 payload-length conditional rate limit**
Traffic can be rate limited based on the IPv4 packet length and IPv6 payload length by specifying a packet-length value or payload-length value or range within the rate-limit filter action. The IPv6 payload-length field does not account for the size of the fixed IP header, which is 40 bytes.

This filter action is supported on ingress IPv4 and IPv6 filter policies only and cannot be configured on egress access or network interfaces.

This rate-limit condition is part of a filter entry action evaluation, and not a filter entry match evaluation. It is checked after the packet is determined to match the entry based on the configured filter entry match criteria.

Packets that match a filter policy's entry match criteria and the rate-limit packet-length value or rate-limit payload-length value are rate limited. Packets that match only the filter policy's entry match criteria and do not match the rate-limit packet-length value or rate-limit payload-length value are forwarded with no further match in subsequent filter entries.

Cflowd, logging, and mirroring apply to all traffic matching the ACL entry regardless of the outcome of the rate limiter and regardless of the packet-length value or payload-length value.

– **IPv4 TTL and IPv6 hop-limit conditional rate limit**

Traffic can be rate limited based on the IPv4 TTL or IPv6 hop-limit by specifying a TTL or hop-limit value or range within the rate-limit filter action using ingress IPv4 or IPv6 filter policies.

The match condition is part of action evaluation (for example, after the packet is determined to match the entry based on other match criteria configured). Packets that match a filter policy entry match criteria and the **rate-limit ttl** or *hop-limit* value are rate limited. Packets that match only the filter policy entry match criteria and do not match the **rate-limit** *ttl* or *hop-limit* value are forwarded with no further matching in the subsequent filter entries.

Cflowd, logging, and mirroring apply to all traffic matching the ACL entry regardless of the outcome of the **rate-limit** value and the *ttl-value* or *hop-limit-value*.

– **pattern conditional rate limit**

Traffic can be rate limited when it is based on a pattern found in the packet header or data payload. The pattern is defined by an expression, mask, offset type, and offset value match in the first 256 bytes of a packet. The pattern expression is up to 8 bytes long. The **offset-type** command identifies the starting point for the offset value and the supported offset-type command options are:

- **layer-3**: layer 3 IP header
- **layer-4**: layer 4 protocol header
- **data**: data payload for TCP or UDP protocols
- **dns-qtype**: DNS request or response query type

The content of the packet is compared with the expression/mask value found at the offset-type command option and offset value defined in the filter entry. For example, if the pattern is expression 0xAA11, mask 0xFFFF, offset-type data, and offset value 20, then the filter entry compares the content of the first 2 bytes in the packet data payload found 20 bytes after the TCP/UDP header with 0xAA11.

This rate limit condition is a filter entry action evaluation, and not a filter entry match evaluation. Within this evaluation, the condition is checked after the packet matches the entry based on the specified filter entry match criteria.

Packets that match a filter policy's entry match criteria and the pattern, are rate limited. Packets that match only the filter policy's entry match criteria and do not match the pattern, are forwarded without a further match in subsequent filter entries.

This filtering capability is supported on ingress IPv4 and IPv6 policies using FP4 and FP5-based line cards and cannot be configured on egress. A filter entry using a pattern is not supported on FP2 or FP3-based line cards. If programmed, the pattern is ignored and the system forwards the packet.

Cflowd, logging, and mirroring apply to all traffic matching this filter entry regardless of the pattern value.

– **extracted traffic conditional rate limit**
Traffic extracted to the CPM can be rate limited using ingress IPv4 and IPv6 filter policies based on filter match criteria. Any IP traffic extracted to the CPM is subject to this filter action, including routing protocols, snooped traffic, and TTL expired traffic.

Packets that match the filter entry match criteria and are extracted to the CPM are rate limited by this filter action and not subject to distributed CPU protection policing.

Packets that match only the filter entry match criteria and are not extracted to the CPM are forwarded with no further match in the subsequent filter entries.

Cflowd, logging, and mirroring apply to all traffic matching the ACL entry regardless of the outcome of the rate limit or the extracted conditional match.

- **Forward Policy-based Routing and Policy-based Forwarding (PBR/PBF) actions**

Allows users to allow ingress traffic but change the regular routing or forwarding that a packet would be subject to. The PBR/PBF is applicable to unicast traffic only. The following PBR or PBF actions are supported (See Configuring filter policies with CLI for more information):

– **egress PBR**

Enabling **egress-pbr** activates a PBR action on egress, while disabling **egress-pbr** activates a PBR action on ingress (default).

The following subset of the PBR actions (defined as follows) can be activated on egress: redirect-policy, next-hop router, and ESI.

Egress PBR is supported in IPv4 and IPv6 filter policies for ESM only. Unicast traffic that is subject to slow-path processing on ingress (for example, IPv4 packets with options or IPv6 packets with hop-by-hop extension header) does not match egress PBR entries. Filter logging, cflowd, and mirror source are mutually exclusive of configuring a filter entry with an egress PBR action. Configuring **pbr-down-action-override**, if supported with a specific PBR ingress action type, is also supported when the action is an egress PBR action. Processing defined by **pbr-down-action-override** does not apply if the action is deployed in the wrong direction. If a packet matches a filter PBR entry and the entry is not activated for the direction in which the filter is deployed, the system forwards the packet. Egress PBR cannot be enabled in system filters.

– **ESI**

Forwards the incoming traffic using VXLAN tunnel resolved using EVPN MP BGP control plane to the first service chain function identified by ESI (Layer 2) or ESI/SF-IP (Layer 3). Supported with VPLS (Layer 2) and IES/VPRN (Layer 3) services. If the service function forwarding cannot be resolved, traffic matches an entry and **action forward** is executed.

For VPLS, no cross-service PBF is supported; that is, the filter specifying ESI PBF entry must be deployed in the VPLS service where BGP EVPN control plane resolution takes place as configured for a specific ESI PBF action. The functionality is supported in filter policies deployed on ingress

VPLS interfaces. BUM traffic that matches a filter entry with ESI PBF is unicast forwarded to the VTEP:VNI resolved through PBF forwarding.

For IES/VPRN, the outgoing R-VPLS interface can be in any VPRN service. The outgoing interface and VPRN service for BGP EVPN control plane resolution must again be configured as part of ESI PBR entry configuration. The functionality is supported in filter policies deployed on ingress IES/VPRN interfaces and in filter policies deployed on ingress and egress for ESM subscribers. Only unicast traffic is subject to ESI PBR; any other traffic matching a filter entry with Layer 3 ESI action is subjected to **action forward**.

When deployed in unsupported direction, traffic matching a filter policy ESI PBR/PBF entry is subject to **action forward**.

– **lsp**

  Forwards the incoming traffic onto the specified LSP. Supports RSVP-TE LSPs (type **static** or **dynamic** only), MPLS-TP LSPs, or SR-TE LSPs. Supported for ingress IPv4/IPv6 filter policies and only deployed on IES SAPs or network interfaces. If the configured LSP is down, traffic matches the entry and **action forward** is executed.

– **mpls-policy**

  Redirects the incoming traffic to the active instance of the MPLS forwarding policy specified by its endpoint. This policy is applicable on any ingress interface (egress is blocked). The traffic is subject to a plain forward if no policy matches the one specified, or if the policy has no programmed instance, or if it is applied on non-L3 interface.

– **next-hop address**

  Changes the IP destination address used in routing from the address in the packet to the address configured in this PBR action. The user can configure whether the next-hop IP address must be direct (local subnet only) or indirect (any IP). In the indirect case, 0.0.0.0 (for IPv4) or :: (for IPv6) is allowed. Default routes may be different per VRF. This functionality is supported for ingress IPv4/IPv6 filter policies only, and is deployed on Layer 3 interfaces.

  If the configured next-hop is not reachable, traffic is dropped and a "ICMP destination unreachable" message is sent. If **indirect** is not specified but the IP address is a remote IP address, traffic is dropped.

– **redirect policy**

  Implements PBR next-hop or PBR next-hop router action with the ability to select and prioritize multiple redirect targets and monitor the specified redirect targets so PBR action can be changed if the selected destination goes down. Supported for ingress IPv4 and IPv6 filter policies deployed on Layer 3 interfaces only. See section Redirect policies for further details.

– **remark DSCP**

  Allows a user to remark the DiffServ Code Points (DSCP) of packets matching filter policy entry criteria. Packets are remarked regardless of QoS-based in- or out-of-profile classification and QoS-based DSCP remarking is overridden. DSCP remarking is supported both as a main action and as an extended action. As a main action, this functionality applies to IPv4 and IPv6 filter policies of any scope and can only be applied at ingress on either access or network interfaces of Layer 3 services only. Although the filter is applied on ingress the DSCP remarking effectively performed on egress. As an extended action, this functionality applies to IPv4 and IPv6 filter policies of any scope and can be applied at ingress on either access or network interfaces of Layer 3 services, or at egress on Layer 3 subscriber interfaces.

– **router**

Changes the routing instance a packet is routed in from the upcoming interface's instance to the routing instance specified in the PBR action (supports both GRT and VPRN redirect). It is supported for ingress IPv4/IPv6 filter policies deployed on Layer 3 interfaces. The action can be combined with the next-hop action specifying direct/indirect IPv4/IPv6 next hop. Packets are dropped if they cannot be routed in the configured routing instance. See section "Traffic Leaking to GRT" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Guide: IES and VPRN* for more information.

– **SAP**

Forwards the incoming traffic onto the specified VPLS SAP. Supported for ingress IPv4/IPv6 and MAC filter policies deployed in VPLS service. The SAP that the traffic is to egress on must be in the same VPLS service as the incoming interface. If the configured SAP is down, traffic is dropped.

– **sdp**

Forwards the incoming traffic onto the specified VPLS SDP. Supported for ingress IPv4/IPv6 and MAC filter policies deployed in VPLS service. The SDP that the traffic is to egress on must be in the same VPLS service as the incoming interface. If the configured SDP is down, traffic is dropped.

– **srte-policy**

Redirects the incoming traffic to the active instance of the SR-TE forwarding policy specified by its endpoint and color. This policy is applicable on any ingress interface (egress is blocked). The traffic is subject to a plain forward if no policy matches the one specified, or if the policy has no programmed instance, or if it is applied on non-Layer 3 interface.

– **srv6-policy**

Redirects the incoming traffic to the active candidate path of the **srv6-policy** identified by its endpoint and color. The user must specify a service SID that exists at the far end. The service SID does not need to be from the same context in which the rule is applied. This rule is applicable on any ingress interface and is blocked on egress. The traffic is subject to a simple forwarding process when:

- no policy matches the specified policy
- the policy has no programmed candidate path
- the policy is applied on a non-Layer 3 interface
- the system performs a simple forward action if no SRv6 source address is configured. No SRv6 source address is configured

– **vprn-target**

Redirects the incoming traffic in a similar manner to combined next-hop and LSP redirection actions, but with greater control and slightly different behavior. This action is supported for both IPv4 and IPv6 filter policies and is applicable on ingress of access interfaces of IES/VPRN services. See Filter policy advanced topics for further details.

Configuring a null endpoint is not blocked but not recommended.

- **ISA forward processing actions**

ISA processing actions allow users to allow ingress traffic and send it for ISA processing as per specified ISA action. See Configuring filter policies with CLI for command details. The following ISA actions are supported:

– **GTP local breakout**

Forwards matching traffic to NAT instead of being GTP tunneled to the mobile user's PGW or GGSN. The action applies to GTP-subscriber-hosts. If filter is deployed on other entities, **action**

**forward** is applied. Supported for IPv4 ingress filter policies only. If ISAs performing NAT are down, traffic is dropped.

- **NAT**

  Forwards matching traffic for NAT. Supported for IPv4/IPv6 filter policies for Layer 3 services in GRT or VPRN. If ISAs performing NAT are down, traffic is dropped.

  For classic CLI options, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Classic CLI Command Reference Guide*.

  For MD-CLI options, see *7450 ESS, 7750 SR, 7950 XRS, and VSR MD-CLI Command Reference Guide*.

- **reassemble**

  Forwards matching packets to the reassembly function. Supported for IPv4 ingress filter policies only. If ISAs performing reassemble are down, traffic is dropped.

- **TCP for MSS adjustment**

  Forwards matching packets (TCP SYN) to an ISA BB group for MSS adjustment. In addition to the IP filter, the user also needs to configure the MSS adjust group under the Layer 3 service to specify the group ID and the new segment-size.

- **HTTP redirect**

  Implements the HTTP redirect captive portal. HTTP GET is forwarded to CPM card for captive portal processing by router. See the HTTP redirect (captive portal) section for more information.

- **ignore match**

  This action allow the user to disable a filter entry, as a result the entry is not programmed in hardware.

In addition to the preceding actions:

- A user can select a default action for a filter policy. The default action is executed on packets subjected to an active filter when none of the filter's active entries matches the packet. By default, filter policies have default action set to drop but the user can select a default action to be forward instead.
- A user can override default action applied to packets matching a PBR/PBF entry when the PBR/PBF target is down using **pbr-down-action-override**. Supported options are to drop the packet, forward the packet, or apply the same action as configured for the filter policy's default action. The override is supported for the following PBR/PBF actions. For the last three actions, the override is supported whether in redundancy mode or not.
  - forward ESI (Layer 2 or Layer 3)
  - forward SAP
  - forward SDP
  - forward next-hop indirect router
  - forward vprn-target

  Table 7: Default behavior when a PBR/PBF target is down defines default behavior for packets matching a PBR/PBF filter entry when a target is down.

*Table 7: Default behavior when a PBR/PBF target is down*

| PBR/PBF action | Default behavior when down |
|---|---|
| Forward esi (any type) | Forward |
| Forward lsp | Forward |
| Forward mpls-policy | Forward |
| Forward next-hop (any type) | Drop |
| Forward redirect-policy | Forward when redirect policy is shutdown |
| Forward redirect-policy | Forward when destination tests are enabled and the best destination is not reachable |
| Forward redirect-policy | Drop when destination tests are not enabled and the best destination is not reachable |
| Forward sap | Drop |
| Forward sdp | Drop |
| Forward srte-policy | Forward |
| Forward router | Drop |
| Forward vprn-target | Forward |

### 4.1.1.6 Viewing filter policy actions

A number of parameters determine the behavior of a packet after it has been matched to a defined criterion or set of criteria:

- the action configured by the user
- the context in which a filter policy is applied. For example, applying a filter policy in an unsupported context can result in simply forwarding the packet instead of applying the configured action.
- external factors, such as the reachability (according to specific test criteria) of a target

Use the following commands to display how a packet is handled by the system.

```
show filter ip
show filter ipv6
show filter mac
```

This section describes the key information displayed as part of the output for the preceding **show** commands, and how to interpret the information.

From a configuration point of view, the **show** command output displays the main action (primary and secondary), as well as the extended action.

The "PBR Target Status" field shows the basic information that the system has of the target based on simple verification methods. This information is only shown for the filter entries which are configured

in redundancy mode (that is, with both primary and secondary main actions configured), and for ESI redirections. Specifically, the target status in the case of redundancy depends on several factors; for example, on a match in the routing table for next-hop redirects, or on VXLAN tunnel resolution for ESI redirects.

The "Downloaded Action" field specifically describes the action that the system performs on the packets that match the criterion (or criteria). This typically depends on the context in which the filter has been applied (whether it is supported or not), but in the case of redundancy, it also depends on the target status. For example, the downloaded action is the secondary main action when the target associated with the primary action is down. In the nominal (for example, non-failure condition) case the "Downloaded Action" reflects the behavior a packet is subject to. However, in transient cases (for example, in the case of a failure) it may not be able to capture what effectively happens to the packet.

The output also displays relevant information such as the default action when the target is down (see Table 7: Default behavior when a PBR/PBF target is down) as well as the overridden default action when **pbr-down-action-override** has been configured.

There are situations where, collectively, this information does not capture what effectively happens to the packet throughout the system. Use the following commands to perform advanced checks and display accurate packet fates.

```
show filter ip effective-action
show filter ipv6 effective-action
show filter mac effective-action
```

The criteria for determining when a target is down. While there is little ambiguity on that aspect when the target is local to the system performing the steering action, ambiguity is much more prominent when the target is distant. Therefore, because the use of **effective-action** triggers advanced tests, a discrepancy is introduced compared to the action when **effective-action** command option is not used. This is, for example, be the case for redundant actions.

#### 4.1.1.7 Filter policy statistics

Filter policies support per-entry, packet/byte match statistics. The cumulative matched packet/Byte counters are available per ingress and per egress direction. Every packet arriving on an interface/service/subscriber using a filter policy increments ingress or egress (as applicable) matched packet/Byte count for a filter entry the packet matches (if any) on the line card the packet ingresses/egresses. For each policy, the counters for all entries are collected from all line cards, summarized and made available to an operator.

Filter policies applied on access interfaces are downloaded only to line cards that have interfaces associated with those filter policies. If a filter policy is not downloaded to any line card, the statistics show 0. If a filter policy is being removed from any of the line cards the policy is currently downloaded to (as result of association change or when a filter becomes inactive), the associated statistics are reset to 0.

Downloading a filter policy to a new line card continues incrementing existing statistics.

Operational notes:

Conditional action match criteria filter entries for **ttl**, **hop-limit**, **packet-length**, and **payload-length** support logging and statistics when the condition is met, allowing visibility of filter matched and action executed. If the condition is not met, packets are not logged and statistics against the entry are not incremented.

### 4.1.1.8 Filter policy logging

SR OS supports logging of the information from the packets that match a specific filter policy. Logging is configurable per filter policy entry by specifying preconfigured filter log (**configure filter log**). A filter log can be applied to ACL filters and CPM hardware filters. Users can configure multiple filter logs and specify:

- memory allocated to a filter log destination
- syslog ID for filter log destination
- filter logging summarization
- wrap-around behavior

Notes related to filter log summarization:

- The implementation of the feature applies to filter logs with destination syslog.
- Summarization logging is the collection and summarization of log messages for one specific log ID within a period of time.
- The summarization interval is 100 seconds.
- Upon activation of a summary, a mini-table with source and destination address and count is created for each type (IPv4, IPv6, and MAC).
- Every received log packet (because of filter match) is examined for source or destination address.
- If the log packet (source/destination address) matches a source/destination address entry in the mini-table, from a packet received previously, the summary counter of the matching address is incremented.
- If source or destination address of the log messages does not match an entry already present in the table, the source/destination address is stored in a free entry in the mini-table.
- In case the mini-table has no more free entries, only total counter is incremented.
- At expiry of the summarization interval, the mini-table for each type is flushed to the syslog destination.

#### Operational note

Conditional action match criteria filter entries for TTL, hop limit, packet length, and payload length support logging and statistics when the condition is met, allowing visibility of filter matched and action executed. If the condition is not met, packets are not logged and statistics against the entry are not incremented.

### 4.1.1.9 Filter policy cflowd sampling

Filter policies can be used to control how cflowd sampling is performed on an IP interface. If an IP interface has cflowd sampling enabled, a user can exclude some flows for interface sampling by configuring filter policy rules that match the flows and by disabling interface sampling as part of the filter policy entry configurations. Use the following commands to disable interface sampling:

- **MD-CLI**

```
configure filter ip-filter entry interface-sample false
configure filter ipv6-filter entry interface-sample false
```

- **classic CLI**

```
configure filter ip-filter entry interface-disable-sample
```

```
configure filter ipv6-filter entry interface-disable-sample
```

If an IP interface has cflowd sampling disabled, a user can enable cflowd sampling on a subset of flows by configuring filter policy rules that match the flows and by enabling cflowd sampling as part of the filter policy entry configurations. Use the following commands to enable cflowd sampling on a subset of flows:

- **MD-CLI**

```
configure filter ip-filter entry filter-sample true
configure filter ipv6-filter entry filter-sample true
```

- **classic CLI**

```
configure filter ip-filter entry filter-sample
configure filter ipv6-filter entry filter-sample
```

The preceding cflowd filter sampling behavior is exclusively driven by match criteria. The sampling logic applies regardless of whether an action was executed (including evaluation of conditional action match criteria, for example, packet length or TTL).

### 4.1.1.10 Filter policy management

#### 4.1.1.10.1 Modifying Existing Filter Policy

There are several ways to modify an existing filter policy. A filter policy can be modified through configuration change or can have entries populated through dynamic, policy-controlled dynamic interfaces; for example, RADIUS, OpenFlow, FlowSpec, or Gx. Although in general, SR OS ensures filter resources exist before a filter can be modified, because of the dynamic nature of the policy-controlled interfaces, a configuration that was accepted may not be applied in H/W because of lack of resources. When that happens, an error is raised.

A filter policy can be modified directly—by changing/adding/deleting the existing entry in that filter policy—or indirectly. Examples of indirect change to filter policy include changing embedded filter entry this policy embeds (see the Filter policy scope and embedded filters section) or changing redirect policy this filter policy uses.

Finally, a filter policy deployed on a specific interface can be changed by changing the policy the interface is associated with.

All of the preceding changes can be done in service. A filter policy that is associated with service/interface cannot be deleted unless all associations are removed first.

For a large (complex) filter policy change, it may take a few seconds to load and initiate the filter policy configuration. Filter policy changes are downloaded to line cards immediately; therefore, users should use filter policy copy or transactional CLI to ensure partial policy change is not activated.

#### 4.1.1.10.2 Filter policy copy

Perform bulk operations on filter policies by copying one filter's entries to another filter. Either all entries or a specified entry of the source filter can be selected to copy. When entries are copied, entry order is preserved unless the destination filter's entry ID is selected (applicable to single-entry copy).

### Filter policy copy and renumbering in classic CLI

**Note:** The information applies to classic CLI.

SR OS supports entry copy and entry renumbering operations to assist in filter policy management.

Use the following commands to copy and overwrite filter entries.

```
configure filter copy ip-filter
configure filter copy ipv6-filter
configure filter copy mac-filter
```

The **copy** command allows overwriting of the existing entries in the destination filter by specifying the **overwrite** command option when using the **copy** command. Copy can be used, for example, when creating new policies from existing policies or when modifying an existing filter policy (an existing source policy is copied to a new destination policy, the new destination policy is modified, then the new destination policy is copied back to the source policy with overwrite specified).

Entry renumbering allows you to change the relative order of a filter policy entry by changing the entry ID. Entry renumbering can also be used to move two entries closer together or further apart, thereby creating additional entry space for new entries.

## 4.1.2 Filter policy advanced topics

### 4.1.2.1 Match list for filter policies

The filter match lists **ip-prefix-list**, **ipv6-prefix-list**, **protocol-list**, **port-list**, **ip-packet-length-list**, and **ipv6-packet-length-list** define a list of IP prefixes, IP protocols, TCP-UDP ports, and packet-length values or ranges that can be used as match criteria for line card IP and IPv6 filters. Additionally, **ip-prefix-list**, **ipv6-prefix-list**, and **port-list** can also be used in CPM filters.

A match list simplifies the filter policy configuration with multiple prefixes, protocols, or ports that can be matched in a single filter entry instead of creating an entry for each.

The same match list can be used in one or many filter policies. A change in match list content is automatically propagated across all policies that use that list.

#### 4.1.2.1.1 Apply-path

The router supports the autogeneration of IPv4 and IPv6 prefix list entries for BGP peers which are configured in the base router or in VPRN services. Use the following commands to configure the autogeneration of IPv6 or IPv4 prefix list entries.

```
configure filter match-list ip-prefix-list apply-path
configure filter match-list ipv6-prefix-list apply-path
```

This capability simplifies the management of CPM filters to allow BGP control traffic from trusted configured peers only. By using the **apply-path** filter, the user can:

- specify one or more regex expression matches per match list, including wildcard matches (".*")
- mix auto-generated entries with statically configured entries within a match list

Additional rules are applied when using **apply-path** as follows:

- Operational and administrative states of a specific router configuration are ignored when auto-generating address prefixes.
- Duplicates are not removed when populated by different auto-generation matches and static configuration.
- Configuration fails if auto-generation of an address prefix results in the filter policy resource exhaustion on a filter entry, system, or line card level.

#### 4.1.2.1.2 Prefix-exclude

A prefix can be excluded from an IPv4 or IPv6 prefix list by using the **prefix-exclude** command.

For example, when the user needs to rate-limit traffic to 10.0.0.0/16 with the exception of 10.0.2.0/24, then the following options are available.

By applying **prefix-exclude**, a single IP prefix list with two prefixes is configured:

**Example: MD-CLI**

```
[ex:/configure filter match-list]
A:admin@node-2# info
    ip-prefix-list "list-1" {
        prefix 10.0.0.0/16 { }
        prefix-exclude 10.0.2.0/24 { }
    }
```

**Example: classic CLI**

```
A:node-2>config>filter>match-list# info
----------------------------------------------
        ip-prefix-list "list-1" create
            prefix 10.0.0.0/16
            prefix-exclude 10.0.2.0/24
        exit
----------------------------------------------
```

Without applying **prefix-exclude**, all eight included subnets should be manually configured in the ip-prefix-list. The following example shows the manual configuration of an IP prefix list.

**Example: MD-CLI**

```
[ex:/configure filter match-list]
A:admin@node-2# info
    ip-prefix-list "list-1" {
        prefix 10.0.0.0/16 { }
        prefix 10.0.0.0/23 { }
        prefix 10.0.3.0/24 { }
        prefix 10.0.4.0/22 { }
        prefix 10.0.8.0/21 { }
        prefix 10.0.16.0/20 { }
        prefix 10.0.32.0/19 { }
        prefix 10.0.64.0/18 { }
        prefix 10.0.128.0/17 { }
    }
```

### Example: classic CLI

```
A:node-2>config>filter>match-list# info
----------------------------------------------
            ip-prefix-list "list-1" create
                prefix 10.0.0.0/16
                prefix 10.0.0.0/23
                prefix 10.0.3.0/24
                prefix 10.0.4.0/22
                prefix 10.0.8.0/21
                prefix 10.0.16.0/20
                prefix 10.0.32.0/19
                prefix 10.0.64.0/18
                prefix 10.0.128.0/17
            exit
----------------------------------------------
```

This is a time consuming and error-prone task compared to using the **prefix-exclude** command.

The filter resources, consumed in hardware, are identical between the two configurations.

A filter match-list using **prefix-exclude** is mutually exclusive with **apply-path**, and is not supported as a match criterion in CPM filter.

Configured **prefix-exclude** prefixes are ignored when no overlapping larger subnet is configured in the prefix-list. For example: prefix-exclude 1.1.1.1/24 is ignored if the only included subnet is 10.0.0.0/16.

## 4.1.2.2 Filter policy scope and embedded filters

The system supports four different filter policies:

- scope template
- scope exclusive
- scope embedded
- scope system

Each scope provides different characteristics and capabilities to deploy a filter policy on a single interface, multiple interfaces or optimize the use of system resources or the management of the filter policies when sharing a common set of filter entries.

### Template and exclusive

A scope template filter policy can be reused across multiple interfaces. This filter policy uses a single set of resources per line card regardless of how many interfaces use it. Template filter policies used on access interfaces consume resources on line cards where the access interfaces are configured only. A scope template filter policy is the most common type of filter policies configured in a router.

A scope exclusive filter policy defines a filter dedicated to a single interface. An exclusive filter allows the highest level of customization but uses the most resources on the system line cards as it cannot be shared with other interfaces.

### Embedded

To simplify the management of filters sharing a common set of filter entries, the user can create a scope embedded filter policy. This filter can then be included in (embedded into) a scope template, scope exclusive, or scope system filter.

Using a scope embedded filter, a common set of filter entries can be updated in a single place and deployed across multiple filter policies. The scope embedded is supported for IPv4 and IPv6 filter policies.

A scope embedded filter policy is not directly downloaded to a line card and cannot be directly referenced in an interface. However, this policy helps the network user provision a common set of rules across different filter policies.

The following rules apply when using a scope embedded filter policy:

- The user explicitly defines the offset at which to insert a filter of scope embedded in a template, exclusive, or system filter. The embedded filter entry-id X becomes entry-id (X + offset) in the main filter.
- Multiple filter scope embedded policies can be included (embedded into) in a single filter policy of scope template, exclusive, or system.
- The same scope embedded filter policy can be included in multiple filter policies of scope template, exclusive, or system.
- Configuration modifications to embedded filter policy entries are automatically applied to all filter policies that embed this filter.
- The system performs a resource management check when a filter policy of scope embedded is updated or embedded in a new filter. If resources are not available, the configuration is rejected. In rare cases, a filter policy resource check may pass but the filter policy can still fail to load because of a resource exhaustion on a line card (for example, when other filter policy entries are dynamically configured by applications like RADIUS in parallel). If that is the case, the embedded filter policy configured is deactivated (configuration is changed from activate to inactivate).
- An embedded filter is never embedded partially in a single filter and resources must exist to embed all the entries in a specific exclusive, template or system filter. However, an embedded filter may be embedded only in a subset of all the filters it is referenced into, only those where there are sufficient resources available.
- Overlapping of filter entries between an embedded filter and a filter of scope template, exclusive or system filter can happen but should be avoided. It is recommended instead that network users use a large enough offset value and an appropriate filter entry-id in the main filter policy to avoid overlapping. In case of overlapping entries, the main filter policy entry overwrites the embedded filter entry.
- Configuring a default action in a filter of scope embedded is not required as this information is not used to embed filter entries.

Figure 22: Embedded Filter Policy shows a configuration with two filter policies of scope template, filter 100 and 200 each embed filter policy 10 at a different offset:

- Filter policy 100 and 200 are of scope template.
- Filter policy 10 of scope embedded is configured with 4 filter entries: entry-id 10, 20, 30, 40.
- Filter policy 100 embed filter 10 at offset 0 and includes two additional static entries with entry-id 20010 and 20020.
- Filter policy 200 embed filter 10 at offset 10000 and includes two additional static entries with entry-id 100 and 110.
- As a result, filter 100 automatically creates entry 10, 20, 30, 40 while filter 200 automatically creates entry 10010, 10020, 10030, 10040. Filter policy 100 and 200 consumed in total 12 entries when both policies are installed in the same line card.

### Example: Scope embedded filter configuration (MD-CLI)

```
[ex:/configure filter]
```

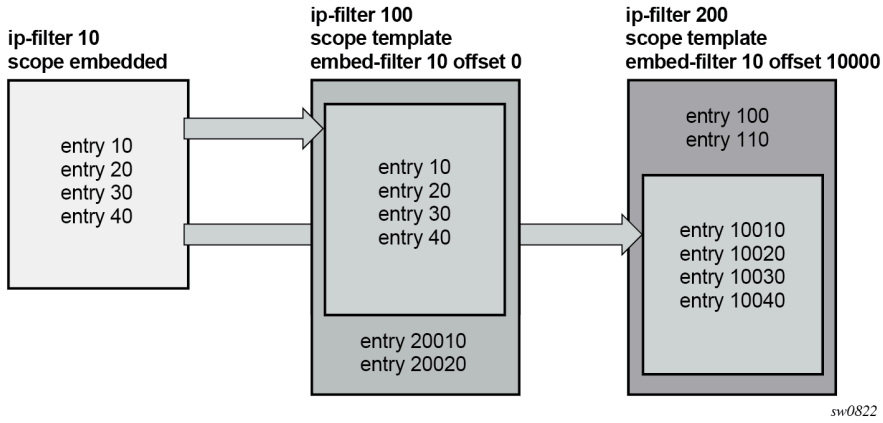
```
A:admin@node-2# info
...
    ip-filter "10" {
        scope embedded
        entry 10 {
        }
        entry 20 {
        }
        entry 30 {
        }
        entry 40 {
        }
    }
    ip-filter "100" {
        scope template
        entry 20010 {
        }
        entry 20020 {
        }
        embed {
            filter "10" offset 0 {
            }
        }
    }
    ip-filter "200" {
        scope template
        entry 100 {
        }
        entry 110 {
        }
        embed {
            filter "10" offset 10000 {
            }
        }
    }
```

### Example: Scope embedded filter configuration (classic CLI)

```
A:node-2>config>filter# info
----------------------------------------------
        ip-filter 10 name "10" create
            scope embedded
            entry 10 create
            exit
            entry 20 create
            exit
            entry 30 create
            exit
            entry 40 create
            exit
        exit
        ip-filter 100 name "100" create
            scope template
            embed-filter 10
            entry 20010 create
            exit
            entry 20020 create
            exit
        exit
        ip-filter 200 name "200" create
            scope template
            embed-filter 10 offset 10000
            entry 100 create
```

```
            exit
            entry 110 create
            exit
        exit
----------------------------------------------
```

*Figure 22: Embedded Filter Policy*




## System

The scope system filter policy provides the most optimized use of hardware resources by programming filter entries after the line cards regardless of how many IPv4 or IPv6 filter policies of scope template or exclusive use this filter. The system filter policy entries are not duplicated inside each policy that uses it, instead, template or exclusive filter policies can be chained to the system filter using the **chain-to-system-filter** command.

When a template of exclusive filter policy is chained to the system filter, system filter rules are evaluated first before any rules of the chaining filter are evaluated (that is chaining filter's rules are only matched against if no system filter match took place).

The system filter policy is intended primarily to deploy a common set of system-level deny rules and infrastructure-level filtering rules to allow, block, or rate limit traffic. Other actions like, for example, PBR actions, or redirect to ISAs should not be used unless the system filter policy is activated only in filters used by services that support such action. The NAT action is not supported and should not be configured. Failure to observe these restrictions can lead to unwanted behavior as system filter actions are not verified against the services the chaining filters are deployed for. System filter policy entries also cannot be the sources of mirroring.

System filter policies can be populated using CLI, SNMP, NETCONF, OpenFlow and FlowSpec. System filter policy entries cannot be populated using RADIUS or Gx.

The following example shows the configuration of an IPv4 system filter:

- System filter policy 10 includes a single entry to rate limit NTP traffic to the Infrastructure subnets.
- Filter policy 100 of scope template is configured to use the system filter using the **chain-to-system-filter** command.

### Example: IPv4 system filter configuration (MD-CLI)

```
[ex:/configure filter]
A:admin@node-2# info
```

```
    ip-filter "10" {
        scope system
        entry 10 {
            description "Rate Limit NTP to the Infrastructure"
            match {
                protocol udp
                dst-ip {
                    ip-prefix-list "Infrastructure IPs"
                }
                dst-port {
                    eq 123
                }
            }
            action {
                accept
                rate-limit {
                    pir 2000
                }
            }
        }
    }
    ip-filter "100" {
        description "Filter scope template for network interfaces"
        chain-to-system-filter true
    }
    system-filter {
        ip "10" { }
    }
```

**Example: IPv4 system filter configuration (classic CLI)**

```
A:node-2>config>filter# info
----------------------------------------------
        ip-filter 10 name "10" create
            scope system
            entry 10 create
                description "Rate Limit NTP to the Infrastructure"
                match protocol udp
                    dst-ip ip-prefix-list "Infrastructure IPs"
                    dst-port eq 123
                exit
                action
                    rate-limit 2000
                exit
            exit
        exit
        ip-filter 100 name "100" create
            chain-to-system-filter
            description "Filter scope template for network interfaces"
        exit
        system-filter
            ip 10
        exit
----------------------------------------------
```

### 4.1.2.3 Filter policy type

The filter policy type defines the list of match criteria available in a filter policy. It provides filtering flexibility by reallocating the CAM in the line card at the filter policy level to filter traffic using additional match criteria not available using filter type normal. The filter type is specific to the filter policy, it is not a system wide or

line card command option. You can configure different filter policy types on different interfaces of the same system and line card.

MAC filter supports three different filter types: normal, ISID, or VID.

IPv4 and IPv6 filters support four different filter types: normal, source MAC, packet length, or destination class.

#### 4.1.2.3.1 IPv4, IPv6 filter type source MAC

This filter policy type provides source MAC match criterion for IPv4 and IPv6 filters.

The following match criteria are not available for filter entries in a source MAC filter policy type:

- **IPv4**

  source IP, DSCP, IP option, option present, multiple option, source-route option

- **IPv6**

  source IP

For a QoS policy assigned to the same service or interface endpoint as a filter policy of type source MAC, QoS IP criteria cannot use source IP or DSCP and QoS IPv6 criteria cannot use source IP.

Filter type source MAC is available for egress filtering on VPLS services only. R-VPLS endpoints are not supported.

Dynamic filter entry embedding using Openflow, FlowSpec and VSD is not supported using this filter type.

#### 4.1.2.3.2 IPv4, IPv6 filter type packet-length

The following match criteria are available using packet-length filter type, in addition to the match criteria that are available using the normal filter type:

- **packet length**

  Total packet length including both the IP header and payload for IPv4 and IPv6 ingress and egress filter policies.

- **TTL or hop limit**

  Match criteria available using FP4-based cards and ingress filter policies; if configured on FP2- or FP3-based cards, the TTL or \hop-limit match criteria part of the filter entries are not programmed in the line card.

The following match criteria are not available for filter entries in a packet-length type filter policy:

- **IPv4**

  DSCP, IP option, option present, multiple option, source-route option

- **IPv6**

  flow label

For a QoS policy assigned to the same service or interface endpoint on egress as a packet-length type filter policy, QoS IP criteria cannot use DSCP match criteria with no restriction to ingress.

This filter type is available for both ingress and egress on all service and router interfaces endpoints with the exception of video ISA, service templates, and PW templates.

Dynamic filter entry embedding using OpenFlow and VSD is not supported using this filter type.

#### 4.1.2.3.3 IPv4, IPv6 filter type destination-class

This filter policy provides BGP destination-class value match criterion capability using egress IPv4 and IPv6 filters, and is supported on network, IES, VPRN, and R-VPLS.

The following match criteria from the normal filter type are not available using the destination-class filter type:

- **IPv4**

  DSCP, IP option, option present, multiple option, source-route option

- **IPv6**

  flow label

Filtering egress on destination class requires the **destination-class-lookup** command to be enabled on the interface that the packet ingresses on. For a QoS policy or filter policy assigned to the same interface, the DSCP remarking action is performed only if a destination-class was not identified for this packet.

System filters, as well as dynamic filter embedding using OpenFlow, FlowSpec, and VSD, are not supported using this filter type.

#### 4.1.2.3.4 IPv4 and IPv6 filter type and embedding

IPv4 and IPv6 filter policy of scope embedded must have the same **type** as the main filter policy of scope template, exclusive or system embedding it:

- If this condition is not met the filter cannot be embedded.
- When embedded, the main filter policy cannot change the filter type if one of the embedded filters is of a different type.
- When embedded, the embedded filter cannot change the filter type if it does not match the main filter policy.

Similarly, the system filter **type** must be identical to the template or exclusive filter to allow chaining when using the **chain-to-system-filter** command.

### 4.1.2.4 Rate limit and shared policer

By default, when a user assigns a filter policy to a LAG endpoint, the system allocates the same user-configured rate limit policer value for each FP of the LAG.

The shared policer feature changes this default behavior. When configured to **true**, the filter policy can only be assigned to endpoints of the same LAG and the configured rate limit policer value is shared between the LAG complexes based on the number of active ports in the LAG on each complex.

The formula to identify the policer value assigned to each FP complex is the following:

- **for same-speed LAGs**

  filter entry rate limit policer value per FP = (configured rate limit) * (number of active ports in the LAG for this FP) / (number of active ports in the LAG)

- **for mixed-speed LAGs and LAGs with a user-configured hash-weight**

  filter entry rate limit policer value per FP = (configured rate limit) * (sum of the weights of all active ports for this FP) / (sum of the weights of all active ports in the LAG)

The shared policer feature is supported for IPv4 and IPv6 filter policies with the template or exclusive scope, in ingress and egress directions.

**Note:**

- Rate limit policer entries embedded in template or exclusive filters follow the shared policer command option from that filter.
- SR-a and VSR do not support the shared policer feature.
- In the MD-CLI configuration combinations with LAG **per-link-hash**, **per-fp-egr-queuing**, **per-fp-sap-instance**, **link-map-profile**, **adapt-qos mode port-fair**, and ESM do not support the shared policer feature.
- In the classic CLI, configuration combinations with LAG **per-link-hash**, **per-fp-egr-queuing**, **per-fp-sap-instance**, **link-map-profile**, **adapt-qos port-fair**, and ESM do not support the shared policer feature.

### 4.1.2.5 Filter policies and dynamic policy-driven interfaces

Filter policy entries can be statically configured using CLI, SNMP, or NETCONF or dynamically created using BGP FlowSpec, OpenFlow, VSD (XMPP), or RADIUS/Diameter for ESM subscribers.

Dynamic filter entries for FlowSpec, OpenFlow, and VSD can be inserted into an IPv4 or IPv6 filter policy. The filter policy must be either exclusive or a template. Additionally, FlowSpec embedding is supported when using a filter policy that defines system-wide filter rules.

#### BGP FlowSpec

BGP FlowSpec routes are associated with a specific routing instance (based on the AFI/SAFI and possibly VRF import policies) and can be used to create filter entries in a filter policy dynamically.

Configure FlowSpec embedding using the following contexts:

- **MD-CLI**

```
configure filter ip-filter embed flowspec
configure filter ipv6-filter embed flowspec
```

- **classic CLI**

```
configure filter ip-filter embed-filter flowspec
configure filter ipv6-filter embed-filter flowspec
```

The following rules apply to FlowSpec embedding:

- The user explicitly defines both the offset at which to insert FlowSpec filter entries and the router instance the FlowSpec routes belong to. The embedded FlowSpec filter entry ID is chosen by the system, in accordance with RFC 5575 *Dissemination of Flow Specification Rules*.

  **Note:** These entry IDs are not necessarily sequential and do not necessarily follow the order at which a rule is received.

- The user can configure the maximum number of FlowSpec filter entries in a specific filter policy at the router or VPRN level using the **ip-filter-max-size** and **ipv6-filter-max-size** commands. This limit defines the boundary for FlowSpec embedding in a filter policy (the offset and maximum number of IPv4 or IPv6 FlowSpec routes).
- When the user configures a template or exclusive filter policy, the router instance defined in the dynamic filter entry for FlowSpec must match the router interface that the filter policy is applied to.
- When using a filter policy that defines system-wide rules, embedding FlowSpec entries from different router instances is allowed and can be applied to any router interfaces.
- See section IPv4/IPv6 filter policy entry match criteria on embedded filter scope for recommendations on filter entry ID spacing and overlapping of entries.

The following information describes the FlowSpec configuration that follows:

- The maximum number of FlowSpec routes in the base router instance is configured for 50,000 entries using the **ip-filter-max-size** command.
- The filter policy 100 (template) is configured to embed FlowSpec routes from the base router instance at offset 100,000. The offset chosen in this example avoids overlapping with statically defined entries in the same policy. In this case, the statically defined entries can use the entry ID range 1-99999 and 149999-2M for defining static entries before or after the FlowSpec filter entries.

The following example shows the FlowSpec configuration.

### Example: FlowSpec configuration (MD-CLI)

```
[ex:/configure router "Base"]
A:admin@node-2# info
    flowspec {
        ip-filter-max-size 50000
    }

[ex:/configure filter ip-filter "100"]
A:admin@node-2# info
...
    ip-filter "100" {
        embed {
            flowspec offset 100000 {
                router-instance "Base"
            }
        }
    }
```

### Example: FlowSpec configuration (classic CLI)

```
A:node-2>config>router# info
----------------------------------------------
        flowspec
            ip-filter-max-size 50000
        exit
----------------------------------------------
A:7750>config>filter# info
----------------------------------------------
        ip-filter 100 name "100" create
            embed-filter flowspec router "Base" offset 100000
        exit
----------------------------------------------
```

### OpenFlow

The embedded filter infrastructure is used to insert OpenFlow rules into an existing filter policy. See Hybrid OpenFlow switch for more information. Policy-controlled auto-created filters are re-created on system reboot. Policy controlled filter entries are lost on system reboot and need to be reprogrammed.

### VSD

VSD filters are created dynamically using XMPP and managed using a Python script so rules can be inserted into or removed from the correct VSD template or embedded filters. XMPP messages received by the 7750 SR are passed transparently to the Python module to generate the appropriate CLI. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 2 Services and EVPN Guide* for more information about VSD filter provisioning, automation, and Python scripting details.

RADIUS or Diameter for subscriber management:

The user can assign filter policies or filter entries used by a subscriber within a preconfigured filter entry range defined for RADIUS or Diameter. See the *7450 ESS, 7750 SR, and VSR Triple Play Service Delivery Architecture Guide* and filter RADIUS-related commands for more information.

## 4.1.2.6 Primary and secondary filter policy action for PBR/PBF redundancy

In some deployments, users may want to specify a backup PBR/PBF target if the primary target is down. SR OS allows the configuration of a primary action as part of a single filter policy entry. The secondary action can only be configured if a primary action is configured.

Use the commands in the following contexts to configure a primary action.

```
configure filter ip-filter entry action
configure filter ipv6-filter entry action
configure filter mac-filter entry action
```

Use the commands in the following contexts to configure a secondary action.

```
configure filter ip-filter entry action secondary
configure filter ipv6-filter entry action secondary
configure filter mac-filter entry action secondary
```

For Layer 2 PBF redundancy, the user can configure the following redundancy command options. Use the **forward sap**, **forward sdp**, **secondary forward sap**, or **secondary forward sdp** options in the following contexts to configure Layer 2 PBF redundancy.

```
configure filter ip-filter entry action
configure filter ipv6-filter entry action
configure filter mac-filter entry action
```

For Layer 3 PBR redundancy, a user can configure any of the following actions as a primary action and any (either same or different than primary) of the following as a secondary action. Furthermore, none of the command options need to be the same between primary and secondary actions. Although the following commands pertain to IPv4 , similar command options also apply to IPv6.

Use the commands in the following contexts to configure forward actions:

- **MD-CLI**

```
configure filter ip-filter entry action forward next-hop nh-ip-vrf
```

```
configure filter ip-filter entry action forward vprn-target
```

- **classic CLI**

```
configure filter ip-filter entry action forward
```

When primary and secondary actions are configured, PBR/PBF uses the primary action if its target is operationally up, or it uses the secondary action if the primary PBR/PBF target is operationally down. If both targets are down, the default action when the target is down (see Table 7: Default behavior when a PBR/PBF target is down), according to the primary action, is used, unless the **pbr-down-action-override** command is configured.

When PBR/PBF redundancy is configured, the user can use sticky destination functionality for a redundant filter entry. When sticky destination is configured, the functionality mimics that of sticky destination configured for redirect policies.

Use the following commands to configure sticky destination.

```
configure filter ip-filter entry sticky-dest
configure filter ipv6-filter entry sticky-dest
configure filter mac-filter entry sticky-dest
```

Use the following commands to force a switchover from the secondary to the primary action when sticky destination is enabled and secondary action is selected.

```
tools perform filter ip-filter entry activate-primary-action
tools perform filter ipv6-filter entry activate-primary-action
tools perform filter mac-filter entry activate-primary-action
```

Sticky destination can be configured even if no secondary action is configured.

The control plane monitors whether primary and secondary actions can be performed and programs forwarding filter policy to use either the primary or secondary action as required. More generally, the state of PBR/PBF targets is monitored in the following situations:

- when a secondary action is configured
- when sticky destination is configured
- when a **pbr-down-action-override** is configured

Use the following command to display which redundant action is activated or downloaded, including when both PBR and PBF targets are down.

```
show filter ip 10 entry 1000
```

The following example shows the partial output of the command as applicable for PBF redundancy.

### Output example

```
…
Primary Action      : Forward (SAP)
  Next Hop           : 1/1/1
  Service Id          : Not configured
  PBR Target Status : Does not exist
Secondary Action    : Forward (SAP)
  Next Hop          : 1/1/2
  Service Id        : Not configured
  PBR Target Status : Does not exist
  PBR Down Action     : Forward (pbr-down-action-override)
```

```
  Downloaded Action   : None
  Dest. Stickiness    : 1000                        Hold Remain    : 0
```

### 4.1.2.7 Extended action for performing two actions at a time

In some deployment scenarios, for example, to realize service function chaining, users may want to perform a second action in addition to a traffic steering action. SR OS supports this behavior by configuring an extended action for a main action. This functionality is supported for Layer 3 traffic steering (that is, PBR) and specifically for the following main actions:

- forward ESI (Layer 3 version)
- forward LSP
- forward next-hop indirect router
- forward redirect-policy
- forward router
- forward VPRN target

The capability to specify an extended action is also supported in the case of PBR redundancy, for the following actions:

- forward next-hop indirect router
- forward VPRN target BGP next hop

The supported extended action is: **remark dscp**

Use the commands in the following contexts to configure the extended action:

- **MD-CLI**

```
configure filter ip-filter entry action ignore-match
configure filter ip-filter entry action ignore-match
```

- **classic CLI**

```
configure filter ip-filter entry action extended-action
configure filter ipv6-filter entry action extended-action
```

#### Extended Action Restrictions

For forward LSP and for actions supporting redundancy, the extended action is not performed when the PBR target is down. Moreover, a filter policy containing an entry with the extended action remark DSCP is blocked in the following cases:

- if applied on ingress with the egress-PBR flag set
- if applied on egress without the egress-PBR flag set

The latter case includes actions that are not supported on egress (and for which egress PBR cannot be set).

### 4.1.2.8 Advanced VPRN redirection

The VPRN target action provides a resilient redirection capability that enables users to redirect packets to a specific BGP next-hop. The user can select an LSP (RSVP-TE, MPLS-TP, or SR-TE LSP) toward that BGP next-hop. If no LSP is configured, the system uses the tunnel that resolves the BGP next-hop. This can operate for either the incoming instance or a different user-configured instance.

The BGP next-hop may not be the next hop currently programmed in the FIB. From that perspective, this redirection overrides the data plane state. The system therefore searches (in the specified instance) whether an active route from that BGP next-hop exists, and if an active route exists, the system determines the service label associated with that route.

This redirection capability supports any label allocation method (label per VRF, label per next-hop, or label per prefix). Because the steering node is not aware of the allocation method of the target, the user must configure the rule accordingly. If no **adv-prefix** is specified in the forward action, the steering node assumes the label per VRF and selects any active route from the BGP next-hop. If the target node is not operating according to the label per a VRF method, the user must specify an appropriate route prefix for which a service label had been advertised by the target node. This instructs the steering node to use the specified service label associated with that route.

Be aware that the system performs an exact match between the specified IP address and the advertised route.

The user may specify an LSP toward the BGP next-hop. If no LSP is specified, the system uses the MPLS-based tunnel that resolves the next-hop, if any. This redirection does not operate with SRv6 tunnels.

This redirection capability operates whether the service instance is BGP-VPN or EVPN-IFL.

This action is resilient in that it tracks events affecting the redirection at the service level and reacts to those events. The system performs the redirection as long as the system can reach the target BGP next-hop using the correct service label. If the redirection cannot be performed (for example, if no LSP is available, the peer is down, or there is no more specific labeled route), the system switches to the secondary action if one has been defined, otherwise it performs normal forwarding. The system can also be configured to perform drop instead of forward, using **pbr-down-action-override**. A maximum of 8k of unique (3-tuple {**bgp-nh**, **router**, **adv-prefix**}) redirection targets can be tracked.

### 4.1.2.9 Destination MAC rewrite when deploying policy-based forwarding

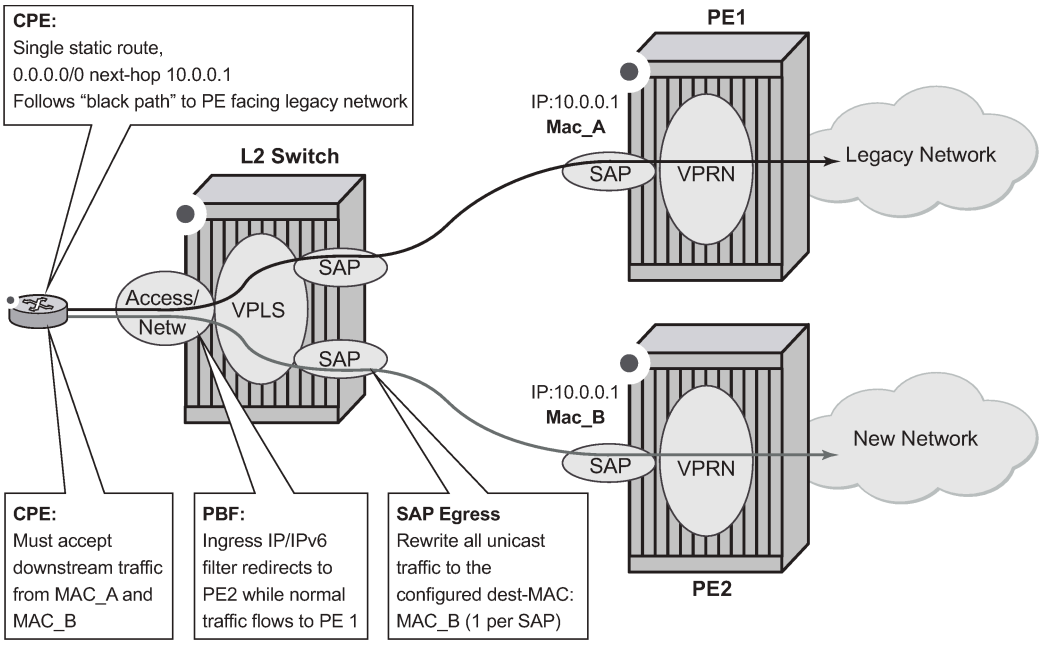
For Layer 2 Policy-Based Forwarding (PBF) redirect actions, a far-end router may discard redirected packets when the PBF changes the destination IP interface the packet arrives on. This happens when a far-end IP interface uses a different MAC address than the IP interface reachable via normal forwarding (for example, one of the routers does not support a configurable MAC address per IP interface).

Use the following command to avoid the discards and deploy egress destination MAC rewrite functionality for VPLS SAPs.

```
configure service vpls sap egress dest-mac-rewrite
```

Figure 23: Layer 2 policy-based forwarding (PBF) redirect action shows a deployment.

*Figure 23: Layer 2 policy-based forwarding (PBF) redirect action*




When enabled, all unicast packets have their destination MAC rewritten to the user-configured value on a Layer 2 switch VPLS SAP. Multicast and broadcast packets are unaffected. The feature:

- Is supported for regular and split-horizon group Ethernet SAPs in a regular VPLS Service
- Is expected to be deployed on a SAP that faces far-end IP interface (either a SAP that is the target of PBF action, as shown in Figure 23: Layer 2 policy-based forwarding (PBF) redirect action, or a VPLS SAP of a downstream Layer 2 switch that is connected to a far-end router—not shown).
- Applies to any unicast egress traffic including LI and mirror.

**Restrictions:**

The following command and the SAP MAC ingress and egress loopback feature are mutually exclusive:

- **MD-CLI**

```
tools perform service id loopback eth sap mac-swap
```

- **classic CLI**

```
tools perform service id loopback eth sap start mac-swap
```

## 4.1.2.10 Network port VPRN filter policy

The network port Layer 3 service-aware filter feature allows users to deploy VPRN service aware ingress filtering on network ports. A single ingress filter of scope template can each be defined for IPv4 and for IPv6 against a VPRN service. The filter applies to all unicast traffic arriving on auto-bind and explicit-spoke network interfaces for that service. The network interface can be either Inter-AS, or Intra-AS. The filter

does not apply to traffic arriving on access interfaces (SAP, spoke SDP, network-ingress (CsC, rVPLS, eVPN).

The same filter can be used on access interfaces of the specific VPRN, can embed other filters (including OpenFlow), can be chained to a system filter, and can be used by other Layer 2 or Layer 3 services.

The filter is deployed on all line cards (chassis network mode D is required). There are no limitations related to filter match/action criteria or embedding. The filter is programmed online cards against ILM entries for this service. All label-types are supported. If an ILM entry has a filter index programmed, that filter is used when the ILM is used in packet forwarding; otherwise, no filter is used on the service traffic.

**Restrictions**

Network port Layer 3 service-aware filters do not support FlowSpec and LI (cannot use filter inside LI infrastructure nor have LI sources within the VPRN filter).

### 4.1.2.11 ISID MAC filters

ISID filters are a type of MAC filter that allows filtering based on the ISID values instead of Layer 2 criteria used by MAC filters of type normal or VID. ISID filters can be deployed on iVPLS PBB SAPs and Epipe PBB SAPs in the following scenarios.

The MMRP usage of the MRP policy ensures automatically that traffic using Group B-MAC is not flooded between domains. However, there could be small transitory periods when traffic originated from PBB BEB with unicast B-MAC destination may be flooded in the BVPLS context as unknown unicast in the BVPLS context for both IVPLS and PBB Epipe. To restrict distribution of this traffic for local PBB services, ISID filters can be deployed. The MAC filter configured with ISID match criterion can be applied to the same interconnect endpoints (BVPLS SAP or PW) as the MRP policy to restrict the egress transmission of any type of frames that contains a local ISID. The ISID filters are applied as required on a per B-SAP or B-PW basis, only in the egress direction.

The ISID match criteria are exclusive with any other criteria under **mac-filter**. A new mac-filter type attribute is defined to control the use of ISID match criteria and must be set to ISID to allow the use of ISID match criteria.

### 4.1.2.12 VID MAC filters

VID filters are a type of MAC filters that extend the capability of current Ethernet ports with null or default SAP tag configuration to match and take action on VID tags. Service delimiting tags (for example, QinQ 1/1/1:10.20 or dot1q 1/1/1:10, where outer tag 10 and inner tags 20 are service delimiting) allow fine granularity control of frame operations based on the VID tag. Service delimiting tags are exact match and are stripped from the frame as shown in Figure 24: VID filtering examples. Exact match or service delimiting tags do not require VID filters. VID filters can only be used to match on frame tags that are after the service delimiting tags.

With VID filters, users can choose to match VID tags for up to two tags on ingress, egress, or both.

- The outer tag is the first tag in the packet that is carried transparently through the service.
- The inner tag is the second tag in the packet that is carried transparently through the service.

VID filters add the capability to perform VID value filter policies on default tags (1/1/1:*, or 1/1/1:x.*, or 1/1/1/:*.0) or null tags (1/1/1, 1/1/1:0, or 1/1/1:x.0). The matching is based on the port configuration and the SAP configuration.

At ingress, the system looks for the two outer-most tags in the frame. If present, any service delimiting tags are removed and not visible to VID MAC filtering. For example:

- 1/1/1:x.y SAP has no tag left for VID MAC filter to match on (outer-tag and inner-tag = 0)
- 1/1/1:x.* SAP has potentially one tag in the * position for VID MAC filter to match on
- SAP such as 1/1/1, 1/1/1:*, or 1/1/1:*.* can have as many as two tags for VID MAC filter to match on
- For the remaining tags, the left (outer-most) tag is what is used as the outer tag in the MAC VID filter. The following tag is used as the inner tag in the filter. If any of these positions do not have tags, a value of 0 is used in the filter. At egress, the VID MAC filter is applied to the frame before adding the additional service tags.

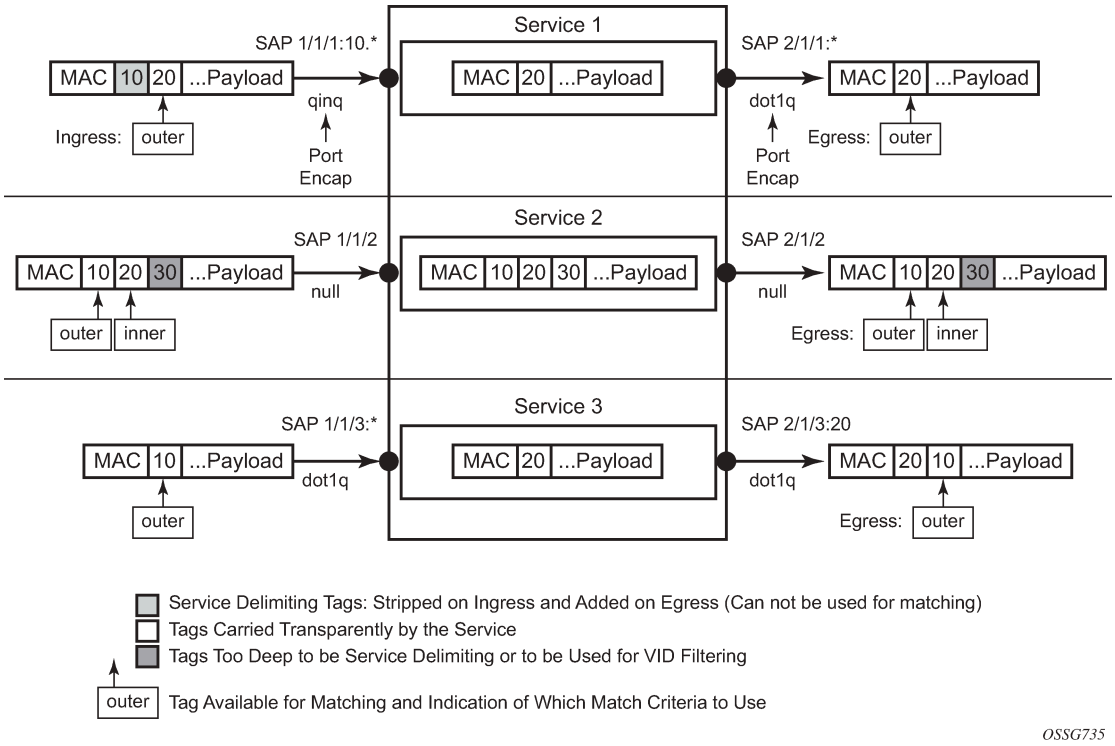
In the industry, the QinQ tags are often referred to as the C-VID (customer VID) and S-VID (service VID). The terms outer tag and inner tag allow flexibility without having to see C-tag and S-tag explicitly. The position of inner and outer tags is relative to the port configuration and SAP configuration. Matching of tags is allowed for up to the first two tags on a frame because service delimiting tags may be 0, 1, or 2 tags.

The meaning of inner and outer has been designed to be consistent for egress and ingress when the number of non-service delimiting tags is consistent. Service 1 in Figure 24: VID filtering examples shows a conversion from QinQ to a single dot1q example where there is one non-service delimiting tag on ingress and egress. Service 2 shows a symmetric example with two non-service delimiting tags (plus and additional tag for illustration) to two non-service delimiting tags on egress. Service 3 shows a single non-service delimiting tag on ingress and two tags with one non-service delimiting tag on ingress and egress.

SAP-ingress QoS command option allows for MAC-criteria type VID, which uses the VID filter matching capabilities of QoS and VID Filters (see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Quality of Service Guide*).

A VID filter entry can also be used as a debug or lawful intercept mirror source entry.

*Figure 24: VID filtering examples*




VID filters are available on Ethernet SAPs for Epipe, VPLS, or I-VPLS including eth-tunnel and eth-ring services.

### 4.1.2.12.1 Arbitrary bit matching of VID filters

In addition to matching an exact value, a VID filter mask allows masking any set of bits. For example: A value of 6 and a mask of 7 would match all VIDs with the lower 3 bits set to 6. VID filters allow explicit matching of VIDs and matching of any bit pattern within the VID tag.

When using VID filters on SAPs, only VID filters are allowed on this SAP. Filters of type normal and ISID are not allowed.

An additional check for the "0" VID tag may be required when using specific wild card operations. For example, frames with no tags on null encapsulated ports match a value of 0 in outer tag and inner tag because there are no tags in the frame for matching. If a zero tag is possible but not wanted, it can be explicitly filtered using exact match on "0" before testing other bits for "0".

Use the following command to configure a special QinQ function for single tagged QinQ frames with a null second tag:

- **MD-CLI**

```
configure service system extended-default-qinq-sap-lookup true
```

- **classic CLI**

```
configure system ethernet new-qinq-untagged-sap
```

Using this command in combination with VID filters is not recommended. The outer tag is the only tag available for filtering on egress for frames arriving from MPLS SDPs or from PBB services, even though additional tags may be carried transparently.

### 4.1.2.12.2 Port group configuration example

*Figure 25: Port groups*

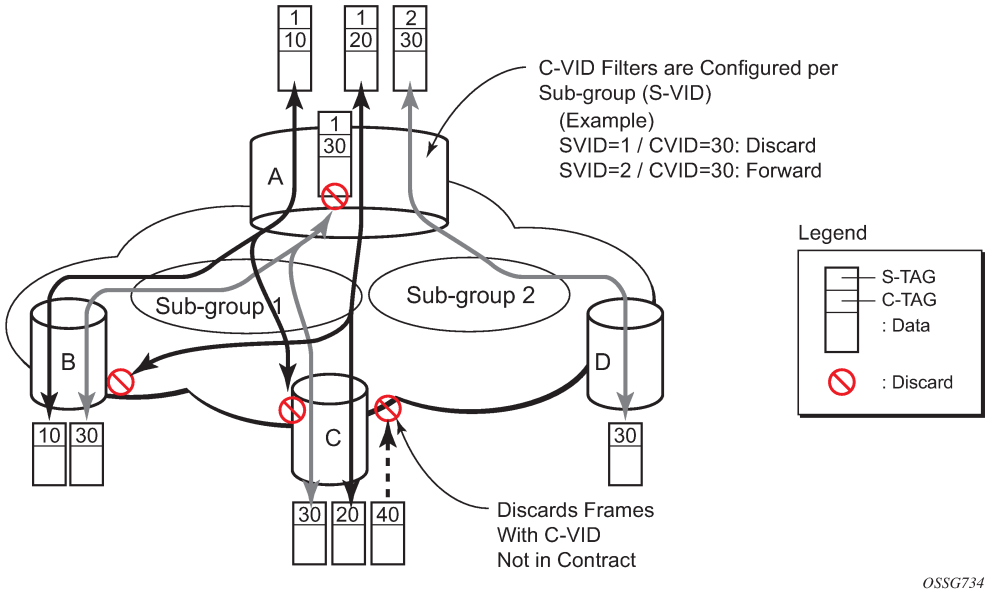



Figure 25: Port groups shows a customer use example where some VLANs are prevented from ingressing or egressing specific ports. In the example, port A sap 1/1/1:1.* would have a filter as shown in the following example, while port A sap 1/1/1:2.* would not.

The following example shows the configuration of the MAC filter command options that apply to port A sap 1/1/1:1.*.

#### Example: MD-CLI

```
[ex:/configure filter]
A:admin@node-2# info
...
    mac-filter "4" {
        default-action accept
        type vid
        entry 1 {
            match {
                outer-tag {
                    tag 30
                    mask 4095
                }
            }
            action {
```

```
                drop
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>filter# info
----------------------------------------------
        mac-filter 4 name "4" create
            default-action forward
            type vid
            entry 1 create
                match frame-type ethernet_II
                    outer-tag 30 4095
                exit
                action drop
            exit
        exit
```

### 4.1.2.13 IP exception filters

IP exception filters scan all outbound traffic entering an NGE domain and allow packets that match the exception filter criteria to transit the NGE domain unencrypted. For information about IP exception filters supported by NGE nodes, see Router encryption exceptions using ACLs.

The most basic IP exception filter policy must have the following:

- an exception filter policy ID
- scope, either exclusive or template
- at least one filter entry with a specified matching criteria

### 4.1.2.14 Redirect policies

SR OS-based routers support configuring of IPv4 and IPv6 redirect policies. Redirect policies allow specifying multiple redirect target destinations and defining status check test methods used to validate the ability for a destination to receive redirected traffic. This destination monitoring allows routers to react to target destination failures. To specify an IPv4 redirect policy, define all destinations to be IPv4. To specify an IPv6 redirect policy, define all destinations to be IPv6. IPv4 redirect policies can only be deployed in IPv4 filter policies. IPv6 redirect policy can only be deployed in IPv6 filter policies.

Redirect policies support the following destination tests:

- **ping-test** – with configurable interval, drop-count, and timeout
- **unicast-rt-test** – for unicast routing reachability, supported only when router instance is configured for a specific redirect policy. The test yields true if the route to the specified destination exists in RTM for the configured router instance.

Each destination is assigned an initial or base priority describing this destination's relative importance within the policy. The destination with the highest priority value is selected as most-preferred destination and programmed online cards in filter policies using this redirect policy as an action. Only destinations that are not disabled by the programmed test (if configured) are considered when selecting the most-preferred destination.

In some deployments, it may not be necessary to switch from a currently active, most-preferred redirect-policy destination when a new more-preferred destination becomes available. Use the following command to enable sticky destination functionality to support such deployments.

```
configure filter redirect-policy sticky-dest
```

When enabled, the currently active destination remains active unless it goes down or a user forces the switch. Use the following command to force the switch.

```
tools perform filter redirect-policy activate-best-dest
```

An optional **sticky-dest** hold-time-up value or **no-hold-time-up** command option is available to delay programming the sticky destination in the redirect policy (transition from **action forward** to PBR action to the most-preferred destination). When the timer is enabled, the first destination that comes up is not programmed and instead the timer is started. After the timer expires, the most-preferred destination at that time is programmed (which may be a different destination from the one that started the timer).

**Note:**

- When the manual switchover to most-preferred destination is executed as described in the preceding information, the hold-time-up is stopped.
- When the timer value is changed, the new value takes immediate effect and the timer is restarted with the new value (or expired if no-hold-time-up is configured).

**Note:**
The **unicast-rt-test** command fails when performed in the context of a VPRN routing instance when the destination is routable only through **grt-leak** functionality. **ping-test** is recommended in these cases.

### Feature restrictions

The following items are feature restrictions:

- Redirect policy is supported for ingress IPv4 and IPv6 filter policies only.
- Different platforms support different scale for redirect policies. Contact your local Nokia representative to ensure the planned deployment does not exceed recommended scale.

## 4.1.2.14.1 Router instance support for redirect policies

There are two modes of deploying redirect policies on VPRN interfaces. The functionality supported depends on the configuration of the redirect-policy router. Use the commands in the following context to configure the redirect-router policy:

- **MD-CLI**

  ```
  configure filter redirect-policy router-instance
  ```

- **classic CLI**

  ```
  configure filter redirect-policy router
  ```

- Redirect policy with router enabled (recommended):

  - When a PBR destination is up, the PBR lookup is performed in the redirect policy's configured routing instance. When that instance differs from the incoming interface where the filter policy using the specific redirect policy is deployed, the PBR action is equivalent to forward next-hop router filter policy action.
  - When all PBR destinations are down (or hardware does not support action router), the system will simply forward the packet and the PBR lookup is performed in the routing instance of the incoming interface where the filter policy using the specific redirect policy is deployed.
  - Any destination tests configured are executed in the routing context specified by the redirect policy.
  - Changing router configuration for a redirect policy brings all destinations with a test configured down. The destinations are brought up after the test confirms reachability based on the new redirect policy router configuration.
- Redirect policy with router disabled or with router not supported (legacy):
  - When a PBR destination is up, the PBR lookup is performed in the routing instance of the incoming interface where the filter policy using the specific redirect policy is deployed.
  - When all PBR destinations are down, the system will simply forward the packet and the PBR lookup is performed in the routing instance of the incoming interface where the filter policy using the specific redirect policy is deployed.
  - Any destination tests configured are always executed in the base router instance regardless of the router instance of the incoming interface where the filter policy using the specific redirect policy is deployed.

**Restrictions**

Only **unicast-rt-test** and **ping-test** are supported when redirect-policy router is enabled.

### 4.1.2.14.2 Binding redirect policies

Redirect policies can switch from a specific destination to a new destination in a coordinated manner as opposed to independently as a function of the reachability test results of their configured destinations. Use the commands in following context to bind together destinations of redirect policies.

```
configure filter redirect-policy-binding
```

SR OS combines the reachability test results (either TRUE or FALSE) from each of the bound destinations and forms a master test result which prevails over each independent result. The combined result can be obtained by applying either an AND function or an OR function. For the AND function, all destinations must be UP (reachability test result equals TRUE) for each destination to be considered UP. Conversely, a single destination must be DOWN for each to be considered DOWN; for the OR case, a single destination needs to be UP for each destination to be considered UP. Apart from the master test, which overrides the test result of each destination forming a binding, redirect policies are unaltered. For stickiness capability, switching toward a more-preferred destination in a specified redirect policy does not occur until the timers (if any) of each of the associated destinations have expired.

There is no specific constraint about destinations that can be bound together. For example, it is possible to bind destinations of different address families (IPv4 or IPv6), destinations with no test, destinations with multiple tests, or destinations of redirect policies which are administratively down. However, some specific scenarios exist when binding redirect policies:

- A destination that is in the Administratively down state is considered DOWN (that is, as if its test result was negative, even if no test had been performed).
- An Administratively down redirect policy is equivalent to a policy with all destinations in an Administratively down state. The system performs a simple forward.
- A destination with no test is considered always UP.
- If a destination has multiple tests, all tests must be positive for the destination to be considered UP (logical AND between its own tests results).
- Destination tests are performed even if a redirect policy has not been applied (that is, not declared as an action of a filter which itself has been applied).

### 4.1.2.15 HTTP redirect (captive portal)

SR OS routers support redirecting HTTP traffic by using the line card ingress IP and IPv6 filter policy action HTTP redirect. This capability is mainly used in the **configure subscriber-mgmt** context to redirect a subscriber web session to a captive portal landing page:

Examples of use cases include redirecting a subscriber after initial connection to a new network to accept the terms of service, or a subscriber out-of-quota redirection.

Traffic matching the HTTP redirect filter entry is sent to the SF/CPM for HTTP redirection:

- The SF/CPM completes the TCP three-way handshake for new TCP sessions on behalf of the intended server, and responds to the HTTP GET request with a 302 redirect. Therefore, the subscriber web session is redirected to the portal landing page configured in the HTTP redirect filter action.
- Non TCP flows are ignored.
- TCP flows other than HTTP, matching an **http-redirect** filter action, are TCP reset after the three-way handshake. Therefore, it is recommended to configure the **http-redirect** filter entry to match only TCP port 80. HTTPs uses TLS as underlying protocol, and cannot be redirected to a landing page.

Additional subscriber information may be required by the captive portal. This information can be appended as variables in the **http-redirect** URL and automatically substituted with the relevant subscriber session data, as follows:

- $IP: subscriber host IP address
- $MAC: subscriber host MAC address
- $URL: original requested URL
- $SAP: subscriber SAP
- $SUB: subscriber identification string
- $CID: circuit-ID, or interface-ID of the subscriber host (hexadecimal format)
- $RID: remote-ID of the subscriber host (hexadecimal format)
- $SAPDESC: configured SAP description

The recommended filter configuration to redirect HTTP traffic page is described in the following information using ingress ip-filter policy "10":

- entry 10: Allows DNS UDP port 53 to a list of allowed DNS servers. Allowing DNS is mandatory for a web client to resolve a URL in the first place. The UDP port directionality indicates DNS request. The destination IP match criteria is optional, creating a list that includes the user DNS, and the most

common open DNS servers provide the most security, allowing, alternatively, UDP -port 53 alone is another option.

- entry 20: Allows HTTP TCP port 80 traffic to the portal landing page defined as a prefix-list. The TCP port directionality indicates an HTTP request. Optionally, the user can create an additional entry to allow TCP port 443 in case the landing page uses both HTTP and HTTPS.
- entry 30: Redirects all TCP port 80 traffic, other than entry 20, to the landing page URL http://www.mydomain/com/redirect.html?subscriber=$SUB&ipaddress=$IP&mac=$MAC&location=$SAP .
- entry 40: Drops explicitly any other IP flows, as in the following configuration example.

### Example: Redirect HTTP filter configuration (MD-CLI)

```
[ex:/configure filter]
A:admin@node-2# info
   ip-filter "10" {
        entry 10 {
            description "Allow DNS Traffic to DNS servers"
            match {
                protocol udp
                ip {
                ip-prefix-list "dns-servers"
                }
                dst-port {
                    eq 53
                }
            }
            action {
                accept
            }
        }
        entry 20 {
            description "Allow HTTP traffic to redirect portal"
            match {
                protocol tcp
                ip {
                ip-prefix-list "portal-servers"
                }
                dst-port {
                    eq 80
                }
            }
            action {
                accept
            }
        }
        entry 30 {
            description "HTTP Redirect all other TCP 80 flows"
            match {
                protocol tcp
                dst-port {
                    eq 80
                }
            }
            action {
                http-redirect {
                    url "http://www.mydomain/com/redirect.html?
subscriber=$SUB&ipaddress=$IP&mac=$MAC&location=$SAP."
                }
            }
        }
        entry 40 {
```

```
            description "Drop anything else"
            action {
                drop
            }
        }
    }
```

### Example: Redirect HTTP filter configuration (classic CLI)

```
A:node-2>config>filter# info
----------------------------------------------
        ip-filter 10 name "10" create
            entry 10 create
                description "Allow DNS Traffic to DNS servers"
                match protocol udp
                    dst-ip ip-prefix-list "dns-servers"
                    dst-port eq 53
                exit
                action
                    forward
                exit
            exit
            entry 20 create
                description "Allow HTTP traffic to redirect portal"
                match protocol tcp
                    dst-ip ip-prefix-list "portal-servers"
                    dst-port eq 80
                exit
                action
                    forward
                exit
            exit
            entry 30 create
                description "HTTP Redirect all other TCP 80 flows"
                match protocol tcp
                    dst-port eq 80
                exit
                action
                    http-redirect "http://www.mydomain/com/redirect.html?
subscriber=$SUB&ipaddress=$IP&mac=$MAC&location=$SAP."
                exit
            exit
            entry 40 create
                description "Drop anything else"
                action
                    drop
                exit
            exit
        exit
----------------------------------------------
```

Also, the router supports two redirect scale modes that are configurable at the system level. The **optimized-mode** improves the number of HTTP redirect sessions supported by system as compared to if optimized mode is disabled.

### Example: Optimized-mode configuration (MD-CLI)

```
[ex:/configure system cpm-http-redirect]
A:admin@node-2# info detail
 ...
    optimized-mode true
```

### Example: Optimized-mode configuration (classic CLI)

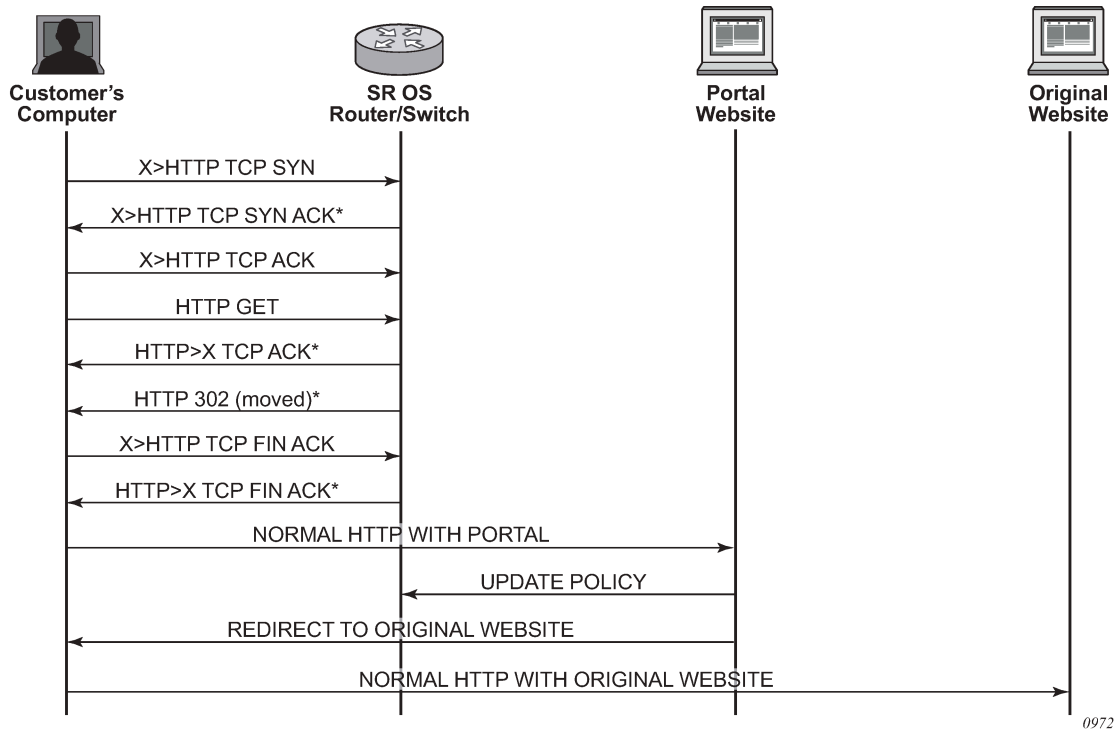
```
A:node-2>config>system>cpm-http-redirect# info detail
----------------------------------------------
            optimized-mode
----------------------------------------------
```

## 4.1.2.15.1 Traffic flow

The following example provides a brief scenario of a subscriber connecting to a new network, where it is required to authenticate or accept the network terms of use, before getting access to Internet:

1. The subscriber typically receives an IP address upon connecting to the network using DHCP, and is assigned a filter policy to redirect HTTP traffic to a web portal.
2. The subscriber HTTP session TCP traffic is intercepted by the router. The CPM completes the TCP three-way handshake on behalf of the destination HTTP server, and responds to the HTTP request with an HTTP 302 "Moved Temporarily" response. This response contains the URL of the web portal configured in the filter policy.
3. Upon receiving this redirect message, the subscriber web browser closes the original TCP session, and opens a new TCP session to the redirection portal.
4. The subscriber can now authenticate or accept the terms of use. After, the subscriber filter policy is dynamically modified.
5. The subscriber can now connect to the original Internet site.

*Figure 26: Web redirect traffic flow*



### 4.1.2.16 Filter policy-based ESM service chaining

In some deployments, users may select to redirect ESM subscribers to Value Added Services (VAS). Various deployment models can be used but often subscribers are assigned to a specific residential tier-of-service, which also defines the VAS available to subscribers of the specific tier. The subscribers are redirected to VAS based on tier-of-service rules, but such an approach can be hard to manage when many VAS services/tiers of service are needed. Often the only way to identify a subscriber's traffic with a specific tier-of-service is to preallocate IP/IPv6 address pools to a specific service tier and use those addresses in VAS PBR match criteria. This creates an application-services to network infrastructure dependency that can be hard to overcome, especially if fast and flexible application service delivery is needed.

Filter policy-based ESM service chaining removes ESM VAS steering to network infrastructure inter-dependency. A user can configure per tier of service or per individual VAS service upstream and downstream service chaining rules without a need to define subscriber or tier-of-service match conditions. Figure 27: ACL filter modeling for ESM service chaining shows a possible ACL model (embedded filters are used for VAS service chaining rules).

On the left in Figure 27: ACL filter modeling for ESM service chaining, the per-tier-of-service ACL model is depicted. Each tier of service (Gold or Silver) has a dedicated embedded VAS filter ("Gold VAS", "Silver VAS") that contains all steering rules for all service chains applicable to the specific tier. Each VAS filter is then embedded by the ACL filter used by a specific tier. A subscriber is subject to VAS service chain rules based on the per-tier ACL assigned to that subscriber (for example, via RADIUS). If a new VAS rule needs to be added, a user must program that rule in all applicable tiers. Upstream and downstream rules can be configured in a single filter (as shown) or can use dedicated ingress and egress filters.

On the right in Figure 27: ACL filter modeling for ESM service chaining, the per-VAS-service ACL model is depicted. Each VAS has a dedicated embedded filter ("VAS 1", "VAS 2", "VAS 3") that contains all steering rules for all service chains applicable to that VAS service. A tier of service is then created by embedding multiple VAS-specific filters: Gold: VAS 1, VAS 2, VAS 3; Silver: VAS 1 and VAS 3. A subscriber is subject to VAS service chain rules based on the per-tier ACL assigned to that subscriber. If a new VAS rule needs to be added, a user needs to program that rule in a single VAS-specific filter only. Again, upstream and downstream rules can be configured in a single filter (as shown) or can use dedicated ingress and egress filters.

*Figure 27: ACL filter modeling for ESM service chaining*

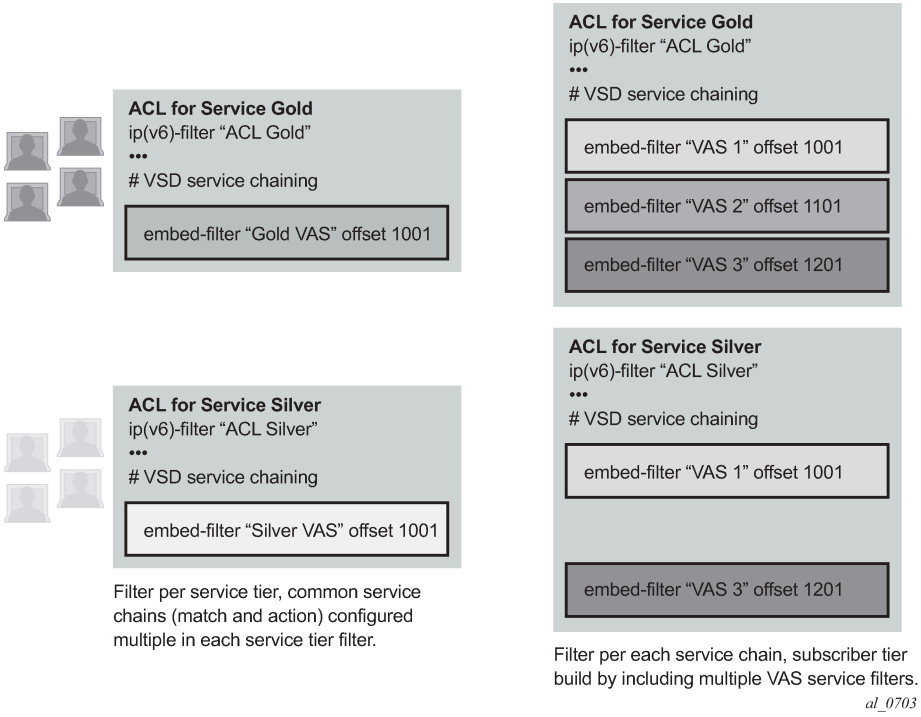



Figure 28: Upstream ESM ACL policy-based service chaining shows upstream VAS service chaining steering using filter policies. Upstream subscriber traffic entering Res-GW is subject to the subscriber's ingress ACL filter assigned to that subscriber by a policy server. If the ACL contains VAS steering rules, the VAS-rule-matching subscriber traffic is steered for VAS processing over a dedicated to-from-access VAS interface in the same or a different routing instance. After the VAS processing, the upstream traffic can be returned to Res-GW by a to-from-network interface (shown in Figure 28: Upstream ESM ACL policy-based service chaining) or can be injected to WAN to be routed toward the final destination (not shown).

*Figure 28: Upstream ESM ACL policy-based service chaining*

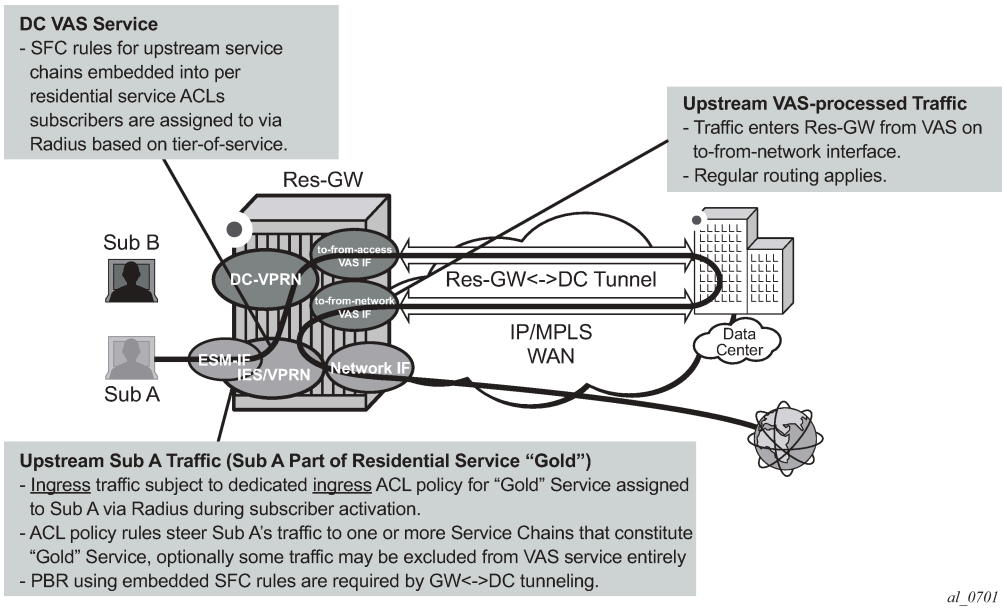
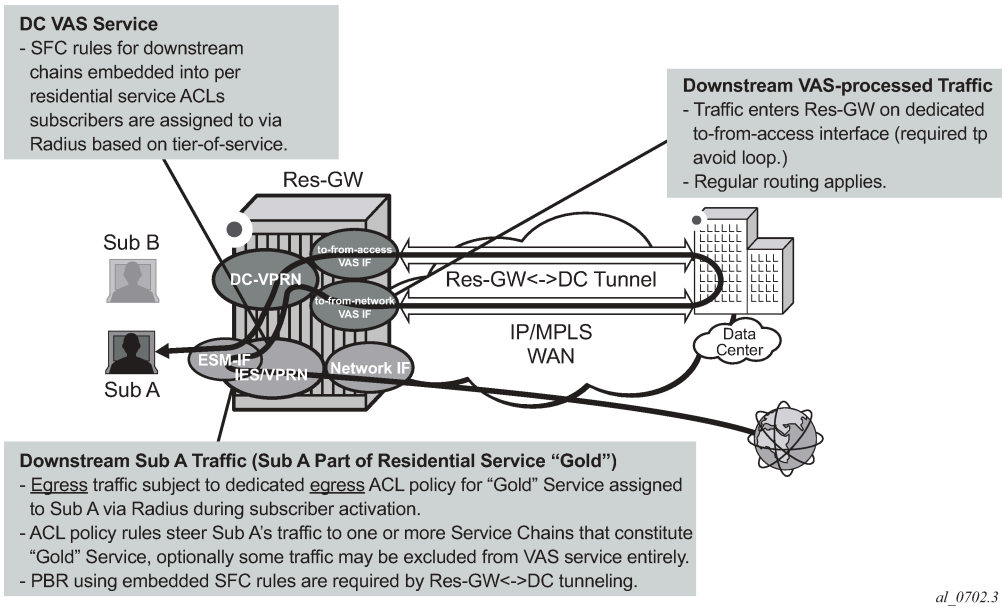



Figure 29: Downstream ESM ACL-policy based service chaining shows downstream VAS service chaining steering using filter policies. Downstream subscriber traffic entering Res-GW is forwarded to a subscriber-facing line card. On that card, the traffic is subject to the subscriber's egress ACL filter policy processing assigned to that subscriber by a policy server. If the ACL contains VAS steering rules, the VAS rule-matching subscriber's traffic is steered for VAS processing over a dedicated to-from-network VAS interface (in the same or a different routing instance). After the VAS processing, the downstream traffic must be returned to Res-GW via a "to-from-network" interface (shown in Figure 29: Downstream ESM ACL-policy based service chaining) to ensure the traffic is not redirected to VAS again when the subscriber-facing line card processes that traffic.

*Figure 29: Downstream ESM ACL-policy based service chaining*




Ensuring the correct configuration for the VAS interface type, for upstream and downstream traffic redirected to a VAS and returned after VAS processing, is critical for achieving loop-free network connectivity for VAS services.

Use the commands in the following contexts to configure the VAS interface type and command options, which are described in the preceding information.

```
configure service vprn if vas-if-type
configure service ies if vas-if-type
configure router if vas-if-type
```

- deployments that use two separate interfaces for VAS connectivity (which is recommended, and required if local subscriber-to-subscriber VAS traffic support is required):
  - **to-from-access**
    - upstream traffic arriving from subscribers over access interfaces must be redirected to a VAS PBR target reachable over this interface for upstream VAS processing
    - downstream traffic destined for subscribers after VAS processing must arrive on this interface, so that the traffic is subject to regular routing but is not subject to Application Assurance diversion, nor to egress subscriber PBR
    - the interface must not be used for downstream pre-VAS traffic; otherwise, routing loops occur
  - **to-from-network**
    - downstream traffic destined for subscribers arriving from network interfaces must be redirected to a VAS PBR target reachable over this interface for downstream VAS processing
    - upstream traffic after VAS processing, if returned to the router, must arrive on this interface so that regular routing can be applied
- deployments that use a single interface for VAS connectivity (optional, no local subscriber-to-subscriber VAS traffic support):

– **to-from-both**
  - both upstream traffic arriving from access interfaces and downstream traffic arriving from the network are redirected to a PBR target reachable over this interface for upstream/downstream VAS processing
  - after VAS processing, traffic must arrive on this interface (optional for upstream), so that the traffic is subject to regular routing but is not subject to AA diversion, nor to egress subscriber PBR
  - the interface must be used for downstream pre-VAS traffic, otherwise, routing loops occur

The ESM filter policy-based service chaining allows users to do the following:

- Steer upstream and downstream traffic per-subscriber with full ACL-flow-defined granularity without the need to specify match conditions that identify subscriber or tier-of-service
- Steer both upstream and downstream traffic on a single Res-GW
- Flexibly assign subscribers to tier-of-service by changing the ACL filter policy a specific subscriber uses
- Flexibly add new services to a subscriber or tier-of-service by adding the subscriber-independent filter rules required to achieve steering
- Achieve isolation of VAS steering from other ACL functions like security through the use of embedded filters
- Deploy integrated Application Assurance (AA) as part of a VAS service chain—both upstream and downstream traffic is processed by AA before a VAS redirect
- Select whether to use IP-Src/IP-Dst address hash or IP-Src/IP-Dst address plus TCP/UDP port hash when LAG/ECMP connectivity to DC is used. Layer 4 inputs are not used in hash with IPv6 packets with extension headers present.

ESM filter policy-based traffic steering supports the following:

- IPv4 and IPv6 steering of unicast traffic using IPv4 and IPv6 ACLs
- use of an action forward redirect-policy filter or an action forward next-hop router filter for IP steering with TCAM-based load-balancing, -to-wire, and sticky destination
- use of an action forward ESI SF-IP VAS interface router filter for an integrated service chaining solution

### Operational notes

The following operational notes apply:

- Downstream traffic steered toward a VAS on the subscriber-facing IOM is reclassified (FC and profile) based on the subscriber egress QoS policy, and is queued toward the VAS based on the network egress QoS configuration. Packets sent toward VAS do not have DSCP remarked (because they are not yet forwarded to a subscriber). DSCP remarking based on subscriber's egress QoS profile only applies to traffic ultimately forwarded to the subscriber (after VAS or not subject to VAS).
- If mirroring of subscriber traffic is configured using ACL entry/subscriber/SAP/port mirror, the mirroring applies to traffic ultimately forwarded to subscriber (after VAS or not subject to VAS). Traffic that is being redirected to VAS cannot be mirrored using an ACL filter implementing PBR action (the same egress ACL filter entry being a mirror source and specifying egress PBR action is not supported).
- Use dedicated ingress and egress filter policies to prevent accidental match of an ingress PBR entry on egress, and the other way around, that results in forwarding or dropping of traffic matching the entry (based on the filter's default action configuration).

### Restrictions

The following restrictions apply:

- This feature is not supported with HSMDAs on subscriber ingress.
- This feature is not supported when the traffic is subject to non-AA ISA on Res-GW.
- Traffic that matches an egress filter entry with an egress PBR action cannot be mirrored, cannot be sampled using cflowd, and cannot be logged using filter logging while being redirected to VAS on a sub-facing line card.
- This feature is not supported with LAC/LNS ESM (PPPoE subscriber traffic encapsulated into or de-encapsulated from L2TP tunnels).
- This feature is not supported for system filter policies.

## 4.1.2.17 Policy-based forwarding for deep packet inspection in VPLS

The purpose policy-based forwarding is to capture traffic from a customer and perform a deep packet inspection (DPI) and forward traffic, if allowed, by the DPI.

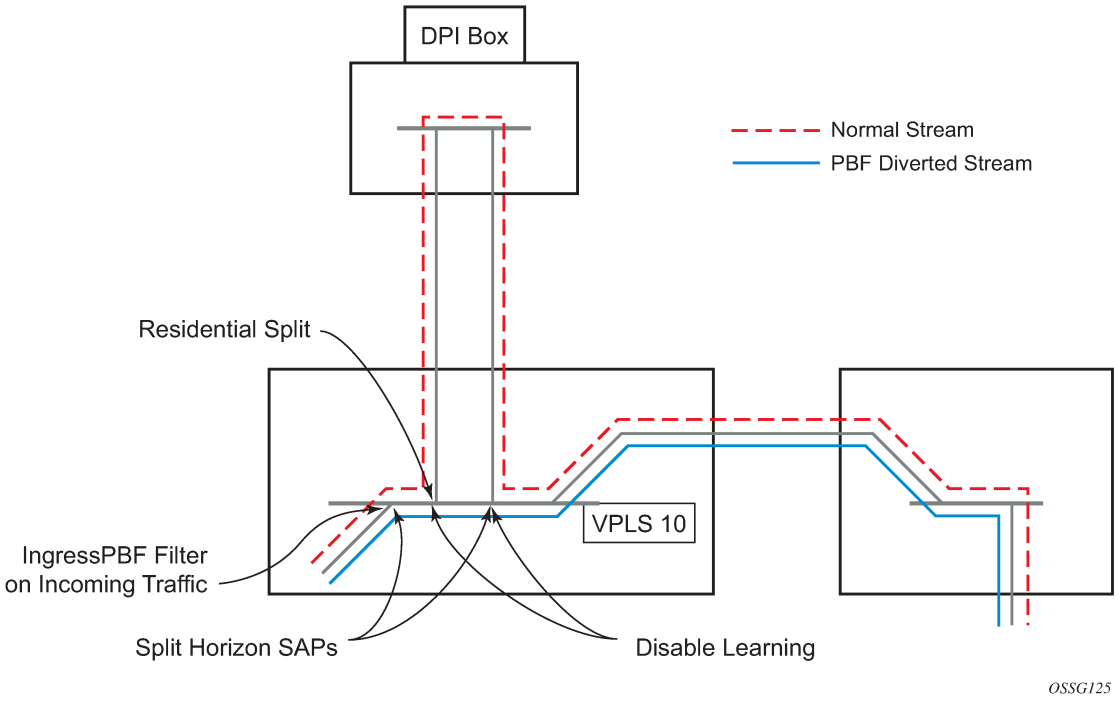
In the following example, the split horizon groups are used to prevent flooding of traffic. Traffic from customers enter at SAP 1/1/5:5. Because mac-filter 100 that is applied on ingress, all traffic with dot1p 07 marking is forwarded to SAP 1/1/22:1, which is the DPI.

DPI performs packet inspection/modification and either drops the traffic or forwards the traffic back into the box through SAP 1/1/21:1. Traffic is then sent to spoke-sdp 3:5.

SAP 1/1/23:5 is configured to see if the VPLS service is flooding all the traffic. If flooding is performed by the router then traffic would also be sent to SAP 1/1/23:5 (which it should not).

Figure 30: Policy-based forwarding for deep packet inspection shows an example to configure policy-based forwarding for deep packet inspection on a VPLS service. For information about configuring services, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 2 Services and EVPN Guide*.

*Figure 30: Policy-based forwarding for deep packet inspection*




### Example: VPLS service configuration with DPI (MD-CLI)

```
[ex:/configure service]
A:admin@node-2# info
...
    vpls "10" {
        admin-state enable
        customer "1"
        service-mtu 1400
        fdb {
            static-mac {
                mac 00:00:00:31:11:01 {
                    sap 1/1/21:1
                }
                mac 00:00:00:31:12:01 {
                    sap 1/1/22:1
                }
                mac 00:00:00:31:13:05 {
                    sap 1/1/23:5
                }
            }
        }
        split-horizon-group "dpi" {
            residential true
        }
        split-horizon-group "split" {
        }
        sap 1/1/21:1 {
            split-horizon-group "split"
            fdb {
                mac-learning {
                    learning false
```

```
                }
            }
        }
        sap 1/1/22:1 {
            split-horizon-group "dpi"
            stp {
                admin-state disable
            }
            fdb {
                mac-learning {
                    learning false
                }
            }
        }
        sap 1/1/23:5 {
        }
    }
```

### Example: VPLS service configuration with DPI (classic CLI)

```
A:node-2>config>service# info
----------------------------------------------
...
        vpls 10 customer 1 create
            service-mtu 1400
            split-horizon-group "dpi" residential-group create
            exit
            split-horizon-group "split" create
            exit
            stp
                shutdown
            exit
            sap 1/1/21:1 split-horizon-group "split" create
                disable-learning
                static-mac 00:00:00:31:11:01 create
            exit
            sap 1/1/22:1 split-horizon-group "dpi" create
                disable-learning
                static-mac 00:00:00:31:12:01 create
            exit
            sap 1/1/23:5 create
                static-mac 00:00:00:31:13:05 create
            exit
            no shutdown
        exit
...
----------------------------------------------
```

### Example: MAC filter configuration (MD-CLI)

```
[ex:/configure filter]
A:admin@node-2# info
    mac-filter "100" {
        default-action accept
        entry 10 {
            log 101
            match {
                dot1p {
                    priority 7
                }
            }
            action {
```

```
                forward {
                    sap {
                        vpls "10"
                        sap-id 1/1/22:1
                    }
                }
            }
        }
    }
...
```

### Example: MAC filter configuration (classic CLI)

```
A:node-2>config>filter# info
----------------------------------------------
...
        mac-filter 100 create
            default-action forward
            entry 10 create
                match
                    dot1p 7 7
                exit
                log 101
                action forward sap 1/1/22:1
            exit
        exit
...
----------------------------------------------
```

### Example: MAC filter added to the VPLS service configuration (MD-CLI)

```
[ex:/configure service]
A:admin@node-2# info
...
    vpls "10" {
        admin-state enable
        customer "1"
        service-mtu 1400
        fdb {
            static-mac {
                mac 00:00:00:31:11:01 {
                    sap 1/1/21:1
                }
                mac 00:00:00:31:12:01 {
                    sap 1/1/22:1
                }
                mac 00:00:00:31:13:05 {
                    sap 1/1/23:5
                }
                mac 00:00:00:31:15:05 {
                    sap 1/1/5:5
                }
            }
        }
        split-horizon-group "dpi" {
            residential true
        }
        split-horizon-group "split" {
        }
        spoke-sdp 3:5 {
        }
        sap 1/1/21:1 {
```

```
            admin-state enable
            split-horizon-group "split"
            fdb {
                mac-learning {
                    learning false
                }
            }
        }
        sap 1/1/22:1 {
            split-horizon-group "dpi"
            stp {
                admin-state disable
            }
            fdb {
                mac-learning {
                    learning false
                }
            }
        }
        sap 1/1/5:5 {
            split-horizon-group "split"
            ingress {
                filter {
                    mac "100"
                }
            }
        }
    }
...
```

**Example: MAC filter added to the VPLS service configuration (classic CLI)**

```
A:node-2>config>service# info
----------------------------------------------
...
        vpls 10 customer 1 create
            service-mtu 1400
            split-horizon-group "dpi" residential-group create
            exit
            split-horizon-group "split" create
            exit
            stp
                shutdown
            exit
            sap 1/1/5:5 split-horizon-group "split" create
                ingress
                    filter mac 100
                exit
                static-mac 00:00:00:31:15:05 create
            exit
            sap 1/1/21:1 split-horizon-group "split" create
                disable-learning
                static-mac 00:00:00:31:11:01 create
            exit
            sap 1/1/22:1 split-horizon-group "dpi" create
                disable-learning
                static-mac 00:00:00:31:12:01 create
            exit
            sap 1/1/23:5 create
                static-mac 00:00:00:31:13:05 create
            exit
            spoke-sdp 3:5 create
            exit
```

```
            no shutdown
        exit
....
-------------------------------------------
```

### 4.1.2.18 Storing filter entries

FP2-, FP3-, FP4, and FP5-based cards store filter policy entries in dedicated memory banks in hardware, also referred to as CAM tables:

- IP/MAC ingress
- IP/MAC egress
- IPv6 ingress
- IPv6 egress

Additional CAM tables for CPM filters are used on SR-1, SR-1s, SR-2s line cards for MAC, IP and IPv6.

#### 4.1.2.18.1 FP4 and FP5-based cards

To optimize both scale and performance, policy entries configured by the operator are compressed by each FP4 and FP5-based line card before being installed in hardware.

This compression can result, in an unexpected scenario typically only achieved in a lab environment, in an overload condition for a specified FP CAM line card. This overload condition can occur when applying a filter policy for the first time on a line card FP or when adding entries to a filter policy.

For a line card ACL filter, the system raises a trap if a specified FP CAM utilization goes beyond 85% utilization.

**Applying a filter policy**

A policy is installed for the first time on a line card FP if no router interface, service interface, SAP, spoke SDP, mesh SDP, or ESM subscriber host was using the policy on this FP.

A policy installed for the first time on a line card FP can lead to a compression failure resulting in an overload condition for this policy on this FP CAM. In this case, none of the entries for the affected filter policy are programmed and traffic is forwarded as if no filter was installed.

**Adding filter entries**

Adding an additional entry to a filter policy can lead to a compression failure resulting in an overload condition.

In this case, the newly added entry is not programmed on the affected FP CAM. Additional entries added to the same policy after the first overload condition are also not programmed on the affected FP CAM as the system attempts to install all outstanding additions in order.

A trap is raised when an overload condition occurs. After the first overload event is detected for a specified ACL FP CAM, the CPM interactively rejects the addition of filter policies or filter entries applied to the same FP CAM, therefore providing an interactive error message to the user or the dynamic provisioning interfaces such as RADIUS.

**Note:**
The filter resource management task on the CPM controls the maximum number of filter entries per FP. If the user attempts to go over the scaling limit, the system returns an interactive error message. This mechanism is independent from the overload state of the FP CAM.

### Removing filter entries

Removing filter entries from a filter policy is always accepted and is used to resolve the overload events.

### Resolving overload

The overload condition should be resolved by the network user before adding new entries or policies in the affected FP CAM.

To identify the affected policy, the system logs the overload event providing slot number, FP number, and impacted CAM. Use the following command to identify policy and policy entries in the system that cannot be programmed on a specific FP CAM.

```
tools dump filter overload
```

To resolve the overload condition, the network user can remove the newly added entries from the affected policy or assign a different policy.

## 4.2 Configuring filter policies with CLI

This section provides information to configure filter policies using the command line interface.

### 4.2.1 Common configuration tasks

This section provides a brief overview of the tasks that must be performed for all IPv4, IPv6, and MAC filter configurations and provides the CLI commands.

#### 4.2.1.1 Creating an IPv4 filter policy

A filter policy has the following attributes:

- policy ID and policy name
- scope: template, exclusive, embedded, system
- type: normal, src-mac, packet-length
- one or more filter entries defining match criteria and action
- default action to define how packets that do not match any of the filter entries are handled

Use the commands in the following context to create a template IPv4 filter policy.

```
configure filter ip-filter
```

#### 4.2.1.1.1 IPv4 filter entry

Within a filter policy, configure filter entries which contain criteria against which ingress, egress traffic is matched. The action specified in the entry determines how the packets are handled, such as drop or forward.

Configure the following to create an IPv4 filter entry:

1. Enter a filter entry ID.
2. Configure the filter matching criteria.
3. Configure the filter action.

The following example shows an IPv4 filter entry configuration.

**Example: MD-CLI**

```
[ex:/configure filter ip-filter "1"]
A:admin@node-2# info
    description "filter-main"
    scope exclusive
    entry 10 {
        description "no-91"
        match {
            src-ip {
                address 10.10.0.100/24
            }
            dst-ip {
                address 10.10.10.91/24
            }
        }
        action {
            drop
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>filter>ip-filter# info
----------------------------------------------
            description "filter-main"
            scope exclusive
            entry 10 create
                description "no-91"
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.0.100/24
                exit
                action drop
            exit
----------------------------------------------
```

##### 4.2.1.1.1.1 Cflowd filter sampling

Within a filter entry, you can specify that traffic matching the associated IPv4 filter entry is sampled if the IPv4 interface is set to cflowd ACL mode. Configuring the **filter-sample** command enables the cflowd tool.

### Example: IPv4 filter entry configuration to sample in cflowd ACL mode (MD-CLI)

```
[ex:/configure filter ip-filter "10"]
A:admin@node-2# info
    description "filter-main"
    scope exclusive
    entry 10 {
        description "no-91"
        filter-sample true
        interface-sample false
        action {
            forward {
                redirect-policy "redirect1"
            }
        }
    }
```

### Example: IPv4 filter entry configuration to sample in cflowd ACL mode (classic CLI)

```
A:node-2>config>filter>ip-filter# info
----------------------------------------------
            description "filter-main"
            scope exclusive
            entry 10 create
                description "no-91"
                filter-sample
                interface-disable-sample
                match
                exit
                action forward redirect-policy redirect1
            exit
----------------------------------------------
```

Within a filter entry, you can also specify that traffic matching the associated IPv4 filter entry is not sampled by cflowd if the IPv4 interface is set to cflowd interface mode.

### Example: IPv4 filter entry configuration not to sample in cflowd ACL mode (MD-CLI)

```
[ex:/configure filter ip-filter "1"]
A:admin@node-2# info
    description "filter-main"
    scope exclusive
    entry 10 {
        description "no-91"
        filter-sample false
        interface-sample true
        match {
        action {
            forward {
                redirect-policy "redirect1"
            }
        }
    }
```

### Example: IPv4 filter entry configuration not to sample in cflowd ACL mode (classic CLI)

```
A:node-2>config>filter>ip-filter# info
----------------------------------------------
            description "filter-main"
            scope exclusive
            entry 10 create
```

```
                description "no-91"
                no filter-sample
                no interface-disable-sample
                match
                exit
                action forward redirect-policy redirect1
            exit
-----------------------------------------------
```

### 4.2.1.2 Creating an IPv6 filter policy

IPv6 filter policy configuration mimics IP filter policy configuration. See Creating an IPv4 filter policy.

### 4.2.1.3 Creating a MAC filter policy

Each filter policy must have the following:

- the filter policy type specified (MAC normal, MAC ISID, MAC VID)
- a filter policy ID
- a default action, either drop or forward
- filter policy scope, either exclusive or template
- at least one filter entry, with a match criterion defined

#### 4.2.1.3.1 MAC filter policy

The following example shows a MAC filter policy configuration.

**Example: MD-CLI**

```
[ex:/configure filter]
A:admin@node-2# info
...
    mac-filter "90" {
        description "filter-west"
        scope exclusive
    }
*[ex:/configure filter]
A:admin@cses-V27# info detail
 ...
    mac-filter "90" {
...
        type normal
...
    }
```

**Example: classic CLI**

```
A:node-2>config>filter# info
----------------------------------------------
...
        mac-filter 90 create
            description "filter-west"
            scope exclusive
```

```
            type normal
        exit
----------------------------------------------
```

#### 4.2.1.3.2 MAC ISID filter policy

The following example shows a MAC ISID filter policy configuration.

**Example: MD-CLI**

```
[ex:/configure filter]
A:admin@node-2# info
    mac-filter "90" {
        description "filter-wan-man"
        scope template
        type isid
        entry 1 {
            description "drop-local-isids"
            match {
                isid {
                    range {
                        start 100
                        end 1000
                    }
                }
            }
            action {
                drop
            }
        }
        entry 2 {
            description "allow-wan-isids"
            match {
                isid {
                    value 150
                }
            }
            action {
                accept
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>filter# info
----------------------------------------------
mac-filter 90 create
     description "filter-wan-man"
     scope template
     type isid
     entry 1 create
          description "drop-local-isids"
          match
               isid 100 to 1000
          exit
          action drop
     exit
     entry 2 create
          description "allow-wan-isids"
```

```
          match
               isid 150
          exit
          action forward
     exit
```

#### 4.2.1.3.3 MAC VID filter policy

The following example shows a MAC VID filter policy configuration.

**Example: MD-CLI**

```
[ex:/configure filter mac-filter "101"]
A:admin@node-2# info
    default-action accept
    type vid
    entry 1 {
        match {
            frame-type ethernet-ii
            outer-tag {
                tag 85
                mask 4095
            }
        }
        action {
            drop
        }
    }
    entry 2 {
        match {
            frame-type ethernet-ii
            outer-tag {
                tag 43
                mask 4095
            }
        }
        action {
            drop
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>filter>mac-filter# info
----------------------------------------------
      default-action forward
      type vid
      entry 1 create
         match frame-type ethernet_II
           outer-tag 85 4095
         exit
         action drop
      exit
      entry 2 create
         match frame-type ethernet_II
           outer-tag 43 4095
         exit
         action drop
      exit
----------------------------------------------
```

#### 4.2.1.3.4 MAC filter entry

Within a filter policy, configure filter entries that contain criteria against which ingress, egress, or network traffic is matched. The action specified in the entry determines how the packets are handled, such as dropping or forwarding.

Configure the following to create a MAC filter entry:

1. Enter a filter entry ID. The system does not dynamically assign a value.
2. Specify matching criteria.
3. Assign an action.

The following example displays a MAC filter entry configuration.

**Example: MD-CLI**

```
[ex:/configure filter]
A:admin@node-2# info
...
    mac-filter "90" {
       entry 1 {
            description "allow-104"
            match
            action {
                drop
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>filter# info
----------------------------------------------
        mac-filter 90 create
            entry 1 create
                description "allow-104"
                match
                exit
                action drop
            exit
        exit
----------------------------------------------
```

### 4.2.1.4 Creating an IPv4 exception filter policy

Configuring and applying IPv4 exception filter policies is optional. Each exception filter policy must have the following:

- an exception filter policy ID
- scope specified, either exclusive or template
- at least one filter entry with matching criteria specified

#### 4.2.1.4.1 IP exception filter policy

Use the commands in the following context to create an IP exception filter policy.

```
configure filter ip-exception
```

The following example displays a template IP exception filter policy configuration.

**Example: MD-CLI**

```
[ex:/configure filter]
A:admin@node-2# info
    ip-exception "1" {
        description "IP-exception"
    }
```

**Example: classic CLI**

```
A:node-2>config>filter# info
----------------------------------------------
...
        ip-exception 1 create
            description "IP-exception"
            scope template
        exit
...
----------------------------------------------
```

#### 4.2.1.4.2 IP exception entry matching criteria

Within an exception filter policy, configure exception entries that contain criteria against which ingress, egress, and network traffic is matched. Packets that match the entry criteria are allowed to transit the NGE domain in cleartext.

Configure the following to create an IP exception entry:

1. Enter an exception filter entry ID. The system does not dynamically assign a value.
2. Specify matching criteria.

Use the commands in the following context to configure the IP exception filter matching criteria.

```
configure filter ip-exception entry match
```

The following example shows an IP exception entry matching criteria configuration.

**Example: MD-CLI**

```
[ex:/configure filter ip-exception "2"]
A:admin@node-2# info
    description "exception-main"
    entry 1 {
        match {
            src-ip {
                address 10.10.10.10/32
            }
            dst-ip {
```

```
                address 10.10.10.91/24
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>filter>ip-except# info
----------------------------------------------
            description "exception-main"
            scope exclusive
            entry 1 create
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.10.10/32
                exit
            exit
----------------------------------------------
```

### 4.2.1.5 Creating an IPv6 exception filter policy

Configuring and applying IPv6 exception filter policies is optional. Each exception filter policy must have the following:

- an exception filter policy ID
- at least one filter entry with matching criteria specified

#### 4.2.1.5.1 IPv6 exception filter policy

**Note:** This information applies for the VSR.

Use the commands in the following context to create an IPv6 exception filter policy.

```
configure filter ipv6-exception
```

The following example shows an IPv6 exception filter policy configuration.

**Example: MD-CLI**

```
*[ex:/configure filter]
A:admin@node-2# info
    ipv6-exception "1" {
        description "IPv6-exception"
    }
```

**Example: classic CLI**

```
*A:node-2>config>filter# info
----------------------------------------------
...
        ipv6-exception 1 create
            description "IPv6-exception"
        exit
...
```

```
-----------------------------------------------
```

#### 4.2.1.5.2 IPv6 exception entry matching criteria

**Note:** This information applies for the VSR.

Within an exception filter policy, configure exception entries that contain criteria against which ingress and network traffic is matched. Packets that match the entry criteria are allowed to transit the IPsec domain in cleartext.

Configure the following to create an IPv6 exception entry:

1. Enter an exception filter entry ID. The system does not dynamically assign a value.
2. Specify matching criteria.

Use the commands in the following context to configure IPv6 exception filter matching criteria.

```
configure filter ipv6-exception entry match
```

The following example shows an IPv6 exception entry matching criteria configuration.

**Example: MD-CLI**

```
[ex:/configure filter ipv6-exception "2"]
A:admin@node-2# info
    description "exception main"
    entry 1 {
        match {
            src-ip {
                address 2001:db8::2/128
            }
            dst-ip {
                address 2001:db8::1/128
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>filter>ipv6-except# info
-----------------------------------------------
            description "exception-main"
            entry 1 create
                match
                    dst-ip 2001:db8::1/128
                    src-ip 2001:db8::2/128
                exit
            exit
-----------------------------------------------
```

### 4.2.1.6 Creating a match list for filter policies

To create a match list, you must:

1. Specify a type of a match list (for example, an IPv4 address prefix list).

2. Define a unique match list name (for example, an IPv4-Deny-List).
3. Specify at least one entry in the list (for example, a valid IPv4 prefix).

The following example shows the IPv4 prefix list configuration and its usage in an IPv4 filter policy.

### Example: MD-CLI

```
[ex:/configure filter]
A:admin@node-2# info
...
    match-list {
        ip-prefix-list "IPv4-Deny-List" {
            description "IPv4-Deny-list"
            prefix 10.0.0.0/21 { }
            prefix 10.254.0.0/24 { }
        }
    }
    ip-filter "ip-edge-filter" {
        scope template
        filter-id 10
        entry 10 {
            match {
                src-ip {
                    ip-prefix-list "IPv4-Deny-List"
                }
            }
            action {
                drop
            }
        }
    }
```

### Example: classic CLI

```
A:node-2>config>filter# info
----------------------------------------------
      match-list
        ip-prefix-list "IPv4-Deny-List" create
           description "IPv4-Deny-list"
           prefix 10.0.0.0/21
           prefix 10.254.0.0/24
        exit
     exit
     ip-filter 10 name "ip-edge-filter" create
        scope template
        entry 10 create
           match
              src-ip ip-prefix-list "IPv4-Deny-List"
           exit
           action
               drop
            exit
      exit
exit

----------------------------------------------
```

### 4.2.1.7 Applying filter policies

Filter policies can be associated with the entities listed in Table 8: Applying filter policies.

*Table 8: Applying filter policies*

| IPv4 and IPv6 Filter Policies | MAC Filter Policies |
|---|---|
| Epipe SAP, spoke SDP | Epipe SAP, spoke SDP |
| spoke SDP | — |
| IES interface SAP, spoke SDP, R-VPLS | — |
| spoke SDP | — |
| VPLS mesh SDP, spoke SDP, SAP | VPLS mesh SDP, spoke SDP, SAP |
| VPRN interface SAP, spoke SDP, R-VPLS, network ingress | — |
| Network interface | — |

#### 4.2.1.7.1 Applying IPv4/IPv6 and MAC filter policies to a service

IP and MAC filter policies are applied by associating them with a SAP or spoke SDP in ingress or egress direction as needed. Filter ID is used to associate an existing filter policy, or if defined, a filter name for that filter policy can be used in the CLI.

**Example: IP and MAC filters assigned to an ingress and egress SAP and spoke SDP (MD-CLI)**

```
[ex:/configure service epipe "5"]
A:admin@node-2# info
    admin-state enable
...
    spoke-sdp 8:8 {
        ingress {
            filter {
                ip "epipe sap default filter"
            }
        }
        egress {
            filter {
                mac "91"
            }
        }
    }
    sap 1/1/1 {
        ingress {
            filter {
                ip "10"
            }
        }
        egress {
            filter {
                mac "92"
```

```
            }
        }
    }
```

### Example: IP and MAC filters assigned to an ingress and egress SAP and spoke SDP (classic CLI)

```
A:node-2>config>service>epipe# info
----------------------------------------------
            sap 1/1/1 create
                ingress
                    filter ip 10
                exit
                egress
                    filter mac 92
                exit
            exit
            spoke-sdp 8:8 create
                ingress
                    filter ip “epipe sap default filter”
                exit
                egress
                    filter mac 91
                exit
            exit
            no shutdown
----------------------------------------------
```

### Example: IPv6 filters assigned to an IES service interface (MD-CLI)

```
[ex:/configure service ies "1001"]
A:admin@node-2# info
    admin-state enable
    customer "1"
    interface "testA" {
        sap 2/1/3:0 {
            ingress {
                filter {
                    ipv6 "100"
                }
            }
            egress {
                filter {
                    ipv6 "100"
                }
            }
        }
        ipv4 {
            primary {
                address 192.22.1.1
                prefix-length 24
            }
        }
        ipv6 {
        }
    }
```

### Example: IPv6 filters assigned to an IES service interface (classic CLI)

```
A:node-2>config>service# info
----------------------------------------------
```

```
        ies 1001 name "1001" customer 1 create
            interface "testA" create
                address 192.22.1.1/24
                ipv6
                exit
                sap 2/1/3:0 create
                    ingress
                        filter ipv6 100
                    exit
                    egress
                        filter ipv6 100
                    exit
                exit
                no shutdown
            exit
-------------------------------------------
```

### 4.2.1.7.2 Applying IPv4/IPv6 filter policies to a network port

IP filter policies can be applied to network IPv4 and IPv6 interfaces. MAC filters cannot be applied to network IP interfaces or to routable IES services. Similar to applying filter policies to service, IPv4/IPv6 filter policies are applied to network interfaces by associating a policy with ingress and egress direction as required. Filter ID is used to associate an existing filter policy, or if defined, a filter name for that filter ID policy can be used in the CLI.

**Example: IP filter applied to an interface at ingress (MD-CLI)**

```
[ex:/configure router "Base"]
A:admin@node-2# info
...
    interface "to-104" {
        port 1/1/1
        egress {
            filter {
                ip "default network egress policy"
            }
        }
        ingress {
            filter {
                ip "10"
            }
        }
        ipv4 {
            primary {
                address 10.0.0.103
                prefix-length 24
            }
        }
    }
...
```

**Example: IP filter applied to an interface at ingress (classic CLI)**

```
A:node-2>config>router# info
#------------------------------------------
# IP Configuration
#------------------------------------------
...
        interface "to-104"
```

```
            address 10.0.0.103/24
            port 1/1/1
            ingress
                filter ip 10
            exit
            egress
                filter ip “default network egress policy”
            exit
        exit
...
#---------------------------------------
```

**Example: IPv4 and IPv6 filters applied to an interface at ingress and egress (MD-CLI)**

```
[ex:/configure router "Base" interface "test1"]
A:admin@node-2# info
    port 1/1/1
    egress {
        filter {
            ip "2"
            ipv6 "1"
        }
    }
    ingress {
        filter {
            ip "2"
            ipv6 "1"
        }
    }
    ipv6 {
        address 3ffe::101:101 {
            prefix-length 120
        }
    }
```

**Example: IPv4 and IPv6 filters applied to an interface at ingress and egress (classic CLI)**

```
A:node-2>config>router>if# info
----------------------------------------------
            port 1/1/1
            ipv6
                address 3FFE::101:101/120
            exit
            ingress
                filter ip 2
                filter ipv6 1
            exit
            egress
                filter ip 2
                filter ipv6 1
            exit
----------------------------------------------
```

### 4.2.1.8 Creating a redirect policy

Configuring and applying redirect policies is optional. Each redirect policy must have the following:

- a destination IP address
- a priority (default is 100)

Configuring a ping test is recommended.

The following example shows the configuration for a redirect policy.

**Example: MD-CLI**

```
[ex:/configure filter]
A:admin@node-2# info
    redirect-policy "redirect1" {
        admin-state enable
        destination 10.10.10.104 {
            priority 105
        }
        destination 10.10.10.105 {
            admin-state enable
            priority 95
            ping-test {
                timeout 30
                drop-count 5
            }
        }
        destination 10.10.10.106 {
            admin-state enable
            priority 90
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>filter# info
----------------------------------------------
        redirect-policy "redirect1" create
            destination 10.10.10.104 create
                priority 105
                exit
                no shutdown
            exit
            destination 10.10.10.105 create
                priority 95
                ping-test
                    timeout 30
                    drop-count 5
                exit
                no shutdown
            exit
            destination 10.10.10.106 create
                priority 90
                exit
                no shutdown
            exit
...
----------------------------------------------
```

### 4.2.1.9 Configuring filter-based GRE tunneling

Traffic matching an IP filter can be tunneled with GRE using the following mechanisms:

- Configure a GRE tunnel template.

- Associate the GRE tunnel template with the forwarding action of an IPv4 or IPv6 filter using the **forward gre-tunnel** command.

The GRE tunnel template defines the command options to create the GRE header used to encapsulate matching IP traffic:

- One or more destination IP addresses must be defined in the GRE tunnel template.
  - If more than one destination is configured, traffic is hashed across all available destinations.
  - GRE-Tunnel-templates using IPv6 transport are limited to a single destination address.
  - Traffic is routed to the selected destination address based on the route table in the forwarding context of the IP filter.
- The source address can be configured to any address and is not validated against a local IP address on the local router.
- The optional GRE key command option can be populated with the ifIndex of the ingress interface on which the matching IP packet was received.
- An optional template command, **skip-ttl-decrement**, allows the TTL of the encapsulated IP packet not to be decremented when encapsulated into the GRE header.

The following example shows the configuration for an IPv4-based GRE tunnel template and an IPv6-based GRE tunnel template.

### Example: MD-CLI

```
[ex:/configure filter]
A:admin@node-2# info
    ip-filter "1" {
        entry 1 {
            pbr-down-action-override forward
            action {
            }
        }
    }
        entry 2 {
            action {
                forward {
                    gre-tunnel "greTunnel_ipv4"
                }
            }
        }
    }
    ip-filter "2" {
        entry 1 {
            action {
                forward {
                    gre-tunnel "greTunnel_ipv6"
                }
            }
        }
    }
    gre-tunnel-template "greTunnel_ipv4" {
        description "10.20.1.5"
        ipv4 {
            source-address 10.20.1.3
            destination-address 9.9.9.9 { }
            destination-address 10.20.1.5 { }
            destination-address 13.13.13.13 { }
        }
    }
```

```
    gre-tunnel-template "greTunnel_ipv6" {
        ipv6 {
            source-address 3ffe::a14:100
            gre-key if-index
            destination-address 3ffe::a01:102 { }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>filter# info
----------------------------------------------
 ...
       gre-tunnel-template "greTunnel_ipv4" create
           description "10.20.1.5"
           ipv4
               source-address 10.20.1.3
               destination-address 9.9.9.9
               destination-address 10.20.1.5
               destination-address 13.13.13.13
           exit
       exit
       gre-tunnel-template "greTunnel_ipv6" create
           ipv6
               gre-key if-index
               source-address 3ffe::a14:100
               destination-address 3ffe::a01:102
           exit
       exit
       ip-filter 1 name "1" create
           entry 1 create
               action
               exit
               pbr-down-action-override forward
           exit
           entry 2 create
               action
                   forward gre-tunnel "greTunnel_ipv4"
               exit
           exit
       exit
       ip-filter 2 name "2" create
           entry 1 create
               action
                   forward gre-tunnel "greTunnel_ipv6"
               exit
           exit
       exit
----------------------------------------------
```

## 4.3 Filter management tasks

This section describes filter policy management tasks.

### 4.3.1 Renumbering filter policy entries

The system exits the matching process when the first match is found and then executes the actions in accordance with the specified action. Because the ordering of entries is important, the numbering sequence may need to be rearranged. Entries should be numbered from the most explicit to the least explicit.

**Example: Renumbering filter policy entries (MD-CLI)**

```
*[ex:/configure filter ip-filter "11"]
A:admin@node-2# rename entry 10 to 15

*[ex:/configure filter ip-filter "11"]
A:admin@node-2# rename entry 20 to 10

*[ex:/configure filter ip-filter "11"]
A:admin@node-2# rename entry 40 to 1
```

**Example: Original filter numbers and updated filter numbers configuration (MD-CLI)**

```
[ex:/configure filter]
A:admin@node-2# info
...
    ip-filter "11" {
        description "filter-main"
        scope exclusive
        entry 10 {
            description "no-91"
            filter-sample true
            interface-sample false
            match {
                src-ip {
                    address 10.10.10.103/24
                }
                dst-ip {
                    address 10.10.10.91/24
                }
            }
            action {
                forward {
                    redirect-policy "redirect1"
                }
            }
        }
        entry 20 {
            match {
                src-ip {
                    address 10.10.0.100/24
                }
                dst-ip {
                    address 10.10.10.91/24
                }
            }
            action {
                drop
            }
        }
        entry 30 {
            match {
                src-ip {
```

```
                    address 10.10.0.200/24
                }
                dst-ip {
                    address 10.10.10.91/24
                }
            }
            action {
                accept
            }
        }
        entry 40 {
            match {
                src-ip {
                    address 10.10.10.106/24
                }
                dst-ip {
                    address 10.10.10.91/24
                }
            }
            action {
                drop
            }
        }
    }
...
-----------------------------------------------
[ex:/configure filter]
A:admin@node-2# info
...
ip-filter "11" {
        description "filter-main"
        scope exclusive
        entry 1 {
            match {
                src-ip {
                    address 10.10.10.106/24
                }
                dst-ip {
                    address 10.10.10.91/24
                }
            }
            action {
                drop
            }
        }
        entry 10 {
            match {
                src-ip {
                    address 10.10.0.100/24
                }
                dst-ip {
                    address 10.10.10.91/24
                }
            }
            action {
                drop
            }
        }
        entry 15 {
            description "no-91"
            filter-sample true
            interface-sample false
            match {
                src-ip {
```

```
                    address 10.10.10.103/24
                }
                dst-ip {
                    address 10.10.10.91/24
                }
            }
            action {
                forward {
                    redirect-policy "redirect1"
                }
            }
        }
        entry 30 {
            match {
                src-ip {
                    address 10.10.0.200/24
                }
                dst-ip {
                    address 10.10.10.91/24
                }
            }
            action {
                accept
            }
        }
    }
...
```

### Example: Renumbering filter policy entries (classic CLI)

```
*A:node-2>config>filter>ip-filter# renum 10 15
*A:node-2>config>filter>ip-filter# renum 20 10
*A:node-2>config>filter>ip-filter# renum 40 1
```

### Example: Original filter numbers and updated filter numbers configuration (classic CLI)

```
A:node-2>config>filter# info
----------------------------------------------
...
        ip-filter 11 create
            description "filter-main"
            scope exclusive
            entry 10 create
                description "no-91"
                filter-sample
                interface-disable-sample
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.10.103/24
                exit
                action forward redirect-policy redirect1
            exit
            entry 20 create
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.0.100/24
                exit
                action drop
            exit
            entry 30 create
                match
                    dst-ip 10.10.10.91/24
```

```
                    src-ip 10.10.0.200/24
                exit
                action forward
            exit
            entry 40 create
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.10.106/24
                exit
                action drop
            exit
        exit
...
-----------------------------------------------
A:node-2>config>filter# info
-----------------------------------------------
...
        ip-filter 11 create
            description "filter-main"
            scope exclusive
            entry 1 create
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.10.106/24
                exit
                action drop
            exit
            entry 10 create
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.0.100/24
                exit
                action drop
            exit
            entry 15 create
                description "no-91"
                filter-sample
                interface-disable-sample
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.10.103/24
                exit
                action forward redirect-policy
                     redirect1
            exit
            entry 30 create
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.0.200/24
                exit
                action forward
            exit
        exit
...
-----------------------------------------------
```

### 4.3.2 Modifying a filter policy

There are several ways to modify an existing filter policy. A filter policy can be modified dynamically as part of subscriber management dynamic insertion or removal of filter policy entries (see the *7450 ESS, 7750 SR, and VSR Triple Play Service Delivery Architecture Guide* for details). A filter policy can be

modified indirectly by configuration change to a match list the filter policy uses (as described earlier in this guide). In addition, a filter policy can be directly edited as described in the following information.

To access a specific IPv4, IPv6, or MAC filter, you must specify the filter ID, or if defined, filter name.

### Example: Modifying a filter (MD-CLI)

In MD-CLI, you can use **delete** to remove a command option from the configuration.

```
*[ex:/configure filter ip-filter "11"]
A:admin@node-2# description "New IP filter info"

*[ex:/configure filter ip-filter "11"]
A:admin@node-2# entry 2

*[ex:/configure filter ip-filter "11" entry 2]
A:admin@node-2# description "new entry"

*[ex:/configure filter ip-filter "11" entry 2]
A:admin@node-2# action drop

*[ex:/configure filter ip-filter "11" entry 2]
A:admin@node-2# match dst-ip address 10.10.10.104/32

*[ex:/configure filter ip-filter "11" entry 2]
A:admin@node-2# exit
```

### Example: Modified IP filter output (MD-CLI)

```
[ex:/configure filter]
A:admin@node-2# info
    ip-filter "11" {
        description "New IP filter info"
        scope exclusive
        entry 1 {
            match {
                src-ip {
                    address 10.10.10.106/24
                }
                dst-ip {
                    address 10.10.10.91/24
                }
            }
            action {
                drop
            }
        }
        entry 2 {
            description "new entry"
            match {
                dst-ip {
                    address 10.10.10.104/32
                }
            }
            action {
                drop
            }
        }
        entry 10 {
            match {
                src-ip {
                    address 10.10.0.100/24
                }
```

```
                dst-ip {
                    address 10.10.10.91/24
                }
            }
            action {
                drop
            }
        }
        entry 15 {
            description "no-91"
            match {
                src-ip {
                    address 10.10.10.103/24
                }
                dst-ip {
                    address 10.10.10.91/24
                }
            }
            action {
                accept
            }
        }
        entry 30 {
            match {
                src-ip {
                    address 10.10.0.200/24
                }
                dst-ip {
                    address 10.10.10.91/24
                }
            }
            action {
                accept
            }
        }
    }
```

### Example: Modifying a filter (classic CLI)

In classic CLI you can use the **no** form of the command to remove the command options or return the command option to the default.

```
*A:node-2>config>filter>ip-filter# description "New IP filter info"
*A:node-2>config>filter>ip-filter# entry 2 create
*A:node-2>config>filter>ip-filter>entry$ description "new entry"
*A:node-2>config>filter>ip-filter>entry# action drop
*A:node-2>config>filter>ip-filter>entry# match dst-ip 10.10.10.104/32
*A:node-2>config>filter>ip-filter>entry# exit
```

### Example: Modified IP filter output (classic CLI)

```
A:node-2>config>filter# info
----------------------------------------------
...
        ip-filter 11 create
            description "New IP filter info"
            scope exclusive
            entry 1 create
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.10.106/24
                exit
```

```
                action drop
            exit
            entry 2 create
                description "new entry"
                match
                    dst-ip 10.10.10.104/32
                exit
                action drop
            exit
            entry 10 create
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.0.100/24
                exit
                action drop
            exit
            entry 15 create
                description "no-91"
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.10.103/24
                exit
                action forward
            exit
            entry 30 create
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.0.200/24
                exit
                action forward
            exit
        exit
...
-----------------------------------------------
```

### 4.3.3 Deleting a filter policy

Before deleting a filter, the filter associations must be removed from all the applied ingress and egress SAPs and network interfaces.

#### Example: Removing the filter from an SAP network interface (MD-CLI)

In MD-CLI, use the **delete** command in all contexts where the filter is used to remove the filter.

```
*[ex:/configure service]
A:admin@node-2# epipe 5

*[ex:/configure service epipe "5"]
A:admin@node-2# sap 1/1/2:3

*[ex:/configure service epipe "5" sap 1/1/2:3]
A:admin@node-2# ingress

*[ex:/configure service epipe "5" sap 1/1/2:3 ingress]
A:admin@node-2# delete filter
```

After you have removed the filter from the SAPs network interfaces, you can delete the filter. The following example shows the deletion of a filter.

### Example: Deleting a filter (MD-CLI)

```
*[ex:/configure filter]
A:admin@node-2# delete ip-filter 11
```

### Example: Removing the filter from an SAP network interface (classic CLI)

In classic CLI, use the **no filter** command in all contexts where the filter is used to remove the filter.

```
*A:node-2>config>service# epipe 5
*A:node-2>config>service>epipe# sap 1/1/2:3
*A:node-2>config>service>epipe>sap# ingress
*A:node-2>config>service>epipe>sap>ingress# no filter
```

After you have removed the filter from the SAPs network interfaces, you can delete the filter. The following example shows the deletion of a filter.

### Example: Deleting a filter (classic CLI)

```
*A:node-2>config>filter# no ip-filter 11
```

## 4.3.4 Modifying a redirect policy

To access a specific redirect policy, the policy name must be specified.

### Example: Modifying a redirect policy (MD-CLI)

Use the **delete** form of the command to remove the command options or return the command option to the default.

```
*[ex:/configure filter]
A:admin@node-2# redirect-policy redirect1

*[ex:/configure filter redirect-policy "redirect1"]
A:admin@node-2# description "New redirect info"

*[ex:/configure filter redirect-policy "redirect1"]
A:admin@node-2# destination 10.10.10.104

*[ex:/configure filter redirect-policy "redirect1" destination 10.10.10.104]
A:admin@node-2# priority 105

*[ex:/configure filter redirect-policy "redirect1" destination 10.10.10.104]
A:admin@node-2# ping-test timeout 20

*[ex:/configure filter redirect-policy "redirect1" destination 10.10.10.104]
A:admin@node-2# ping-test drop-count 7
```

### Example: Modified redirect policy output (MD-CLI)

```
[ex:/configure filter]
A:admin@node-2# info
    redirect-policy "redirect1" {
        admin-state enable
        description "New redirect info"
        destination 10.10.10.104 {
            admin-state enable
```

```
            description "New redirect info"
            priority 105
            ping-test {
                timeout 20
                drop-count 7
            }
        }
        destination 10.10.10.105 {
            admin-state enable
            priority 95
            ping-test {
                timeout 30
                drop-count 5
            }
        }
    }
```

### Example: Modifying a redirect policy (classic CLI)

Use the **no** form of the command to remove the command options or return the command option to the default.

```
*A:node-2>config>filter# redirect-policy redirect1
*A:node-2>config>filter>redirect-policy# description "New redirect info"
*A:node-2>config>filter>redirect-policy# destination 10.10.10.104
*A:node-2>config>filter>redirect-policy>dest# priority 105
*A:node-2>config>filter>redirect-policy>dest# ping-test timeout 20
*A:node-2>config>filter>redirect-policy>dest# ping-test drop-count 7
```

### Example: Modified redirect policy output (classic CLI)

```
A:node-2>config>filter# info
----------------------------------------------
...
        redirect-policy "redirect1" create
            description "New redirect info"
            destination 10.10.10.104 create
                priority 105
                ping-test
                    timeout 20
                    drop-count 7
                exit
                no shutdown
            exit
            destination 10.10.10.105 create
                priority 95
                ping-test
                    timeout 30
                    drop-count 5
                exit
                no shutdown
            exit
            no shutdown
        exit
...
----------------------------------------------
```

### 4.3.5 Deleting a redirect policy

Before a redirect policy can be deleted from the filter configuration, the policy association must be removed from the IP filter.

The following example shows the replacement of redirect policy "redirect1" with redirect policy "redirect2" and the removal of "redirect1" from the filter configuration.

**Example: Replacing and deleting a redirect policy (MD-CLI)**

```
*[ex:/configure filter]
A:admin@node-2# ip-filter 11

*[ex:/configure filter ip-filter "11"]
A:admin@node-2# entry 1

*[ex:/configure filter ip-filter "11" entry 1]
A:admin@node-2# action forward redirect-policy redirect2

*[ex:/configure filter ip-filter "11" entry 1]
A:admin@node-2# exit

*[ex:/configure filter ip-filter "11"]
A:admin@node-2# exit

*[ex:/configure filter]
A:admin@node-2# delete redirect-policy redirect1
```

**Example: Output after deleting a redirect policy (MD-CLI)**

```
[ex:/configure filter ip-filter "11"]
A:admin@node-2# info
    description "This is new"
    scope exclusive
    entry 1 {
        filter-sample true
        interface-sample false
        match {
            src-ip {
                address 10.10.10.106/24
            }
            dst-ip {
                address 10.10.10.91/24
            }
        }
        action {
            forward {
                redirect-policy "redirect2"
            }
        }
    }
    entry 2 {
        description "new entry"
...
```

**Example: Replacing and deleting a redirect policy (classic CLI)**

```
*A:node-2>config>filter# ip-filter 11
*A:node-2>config>filter>ip-filter# entry 1
*A:node-2>config>filter>ip-filter>entry# action forward redirect-policy "redirect2"
```

```
*A:node-2>config>filter>ip-filter>entry# exit
*A:node-2>config>filter>ip-filter# exit
*A:node-2>config>filter# no redirect-policy "redirect1"
```

### Example: Output after deleting a redirect policy (classic CLI)

```
A:node-2>config>filter>ip-filter# info
----------------------------------------------
            description "This is new"
            scope exclusive
            entry 1 create
               filter-sample
               interface-disable-sample
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.10.106/24
                exit
                action forward redirect-policy redirect2
            exit
            entry 2 create
                description "new entry"
...
----------------------------------------------
```

## 4.3.6 Copying filter policies

When changes are to be made to an existing filter policy applied to a one or more SAPs or network interfaces, Nokia recommends to first copy the applied filter policy, then modify the copy and then overwrite the applied policy with the modified copy. This ensures that a policy being modified is not applied when partial changes are done as any filter policy edits are applied immediately to all services where the policy is applied.

New filter policies can also be created by copying an existing policy and renaming the new filter.

The following example displays the copying of the configuration information from an existing IP filter policy "11" to create a new filter policy "12" that can then be edited. After edits are completed, they can be used to overwrite existing policy "11".

### Example: Copying a filter policy (MD-CLI)

```
*[ex:/configure filter]
A:admin@node-2# copy ip-filter 11 to ip-filter 12
```

### Example: Copied filter policy output (MD-CLI)

```
[ex:/configure filter]
A:admin@node-2# info
   ip-filter "11" {
        description "This is new"
        scope exclusive
        entry 1 {
            match {
                src-ip {
                    address 10.10.10.106/24
                }
                dst-ip {
                    address 10.10.10.91/24
                }
```

```
            }
            action {
                drop
                }
            }
        }
        entry 2 {
...
    ip-filter "12" {
        description "This is new"
        scope exclusive
        entry 1 {
            match {
                src-ip {
                    address 10.10.10.106/24
                }
                dst-ip {
                    address 10.10.10.91/24
                }
            }
            action {
                drop
            }
        }
        entry 2 {
...
```

### Example: Copying a filter policy (classic CLI)

```
*A:node-2>config>filter# copy ip-filter 11 to 12
```

### Example: Copied filter policy output (classic CLI)

```
A:node-2>config>filter# info
----------------------------------------------
...
        ip-filter 11 create
            description "This is new"
            scope exclusive
            entry 1 create
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.10.106/24
                exit
                action drop
            exit
            entry 2 create
...
        ip-filter 12 create
            description "This is new"
            scope exclusive
            entry 1 create
                match
                    dst-ip 10.10.10.91/24
                    src-ip 10.10.10.106/24
                exit
                action drop
            exit
            entry 2 create
...
```

# 5 Hybrid OpenFlow switch

## 5.1 Hybrid OpenFlow switching

**Note:** Hybrid OpenFlow Switch (H-OFS) commands are only supported on the VSR.

The H-OFS model allows users to deploy SDN traffic steering using OpenFlow on top of the existing routing and switching infrastructure. Some of the main benefits of the hybrid model include:

- Increased flexibility and speed for new service deployment. The H-OFS model implements flexible, policy-driven, standard-based H-OFS traffic steering that allows deployment of new services and on-demand services through policy updates instead of service and infrastructure programming.
- Evolutionary CAPEX/OPEX-optimized SDN deployment. The H-OFS functionality can be deployed on the existing hardware through software upgrade to realize the benefits of FlexPath programmability. The OpenFlow traffic placement is focused access only (that is, flexible, fast, on-demand service deployment) while network infrastructure provides robustness, resiliency, scale, and security.

In a basic mode of operation, a single OpenFlow Switch instance is configured on the router and controlled by a single OpenFlow controller.

The OF controllers and router exchange OF messages using the OF protocol (version 1.3.1) over the TCP/IP control channel. IPv4 and IPv6 controller addressing are supported. Both out-of-band (default) and in-band management are supported for connectivity to the controller. Transport Layer Security (TLS) is also supported on the control channel. An OF message is processed by the OF switch instance on the router that installs all supported H-OFS traffic steering rules in a flow table for the H-OFS instance. A single table per H-OFS instance is supported.

The H-OFS allows users to:

- Steer IPv4/IPv6 unicast traffic arriving on a Layer 3 interface by programming the 7450 ESS, 7750 SR, 7950 XRS, and VSR L3 PBR ACL actions.
- Steer IPv4/IPv6 unicast traffic arriving on a Layer 2 interface by programming the 7450 ESS, 7750 SR, 7950 XRS, and VSR L2 PBF ACL actions.
- Drop traffic by programming ACL action drop.
- Forward traffic using regular processing by programming ACL action forward.

Steering actions programmed using OpenFlow are functionally equivalent to ACL actions.

The router allows users to control traffic using OF, as follows:

- A user can select a subset of interfaces on the router to have OF rules enabled, by embedding a specific instance of H-OFS in filter policies used only by those interfaces.
- For the interfaces with an H-OFS instance enabled, a user can:
  - Steer all traffic arriving on an interface by programming the flow table with a "match all" entry.
  - Steer a subset of traffic arriving on an interface with this H-OFS instance enabled by programming the flow table with match rules that select a subset of traffic (OpenFlow match criteria are translated to ACL filter match criteria). Unless explicitly listed as a limitation, the SR OS H-OFS supports any

OpenFlow match criteria that can be translated to ACL IPv4/IPv6 filter policy match criteria. A default rule can be assigned for packets that do not match specific rules. These packets can be dropped, forwarded, or sent to the OpenFlow controller.

To enable rules in an H-OFS on an existing service router interface, a user must:

1. Create one or more ingress line card policies.
2. Assign those line card ingress filter policies to the 7450 ESS, 7750 SR, 7950 XRS, and VSR service router interfaces.
3. Embed an H-OFS instance into those line card policies.
4. Program OF rules as required.

OpenFlow can be embedded in IPv4/IPv6 ACL filter policies deployed on:

- Layer 3 IES service interfaces
- Layer 3 network interfaces in base router context
- Layer 3 VPRN service interfaces, including those with NAT
- Layer 2 VPLS service interfaces
- IES/VPRN r-VPLS service interfaces, including those with NAT
- System ACL filters

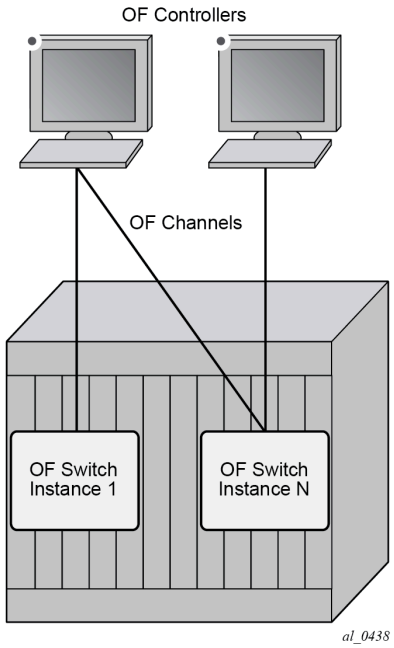
OpenFlow functionality can be enabled with no effect on forwarding performance. Users can move from CLI/SNMP programmed steering rules to OpenFlow operational model in service without service disruption.

The control channel is routed via the GRT, meaning that the controller must be reachable via GRT, or it may be routed via a VPRN. VPRN support requires that a loopback interface corresponding to each OpenFlow switch, reachable via the VPRN, is configured in the VPRN. Then, the VPRN service ID or name and the corresponding OpenFlow control channel loopback address are specified in the OpenFlow switch control channel configuration.

### 5.1.1 Redundant controllers and multiple switch instances

The user can configure one or more instances of an H-OFS (using SNMP or CLI interfaces) with each instance controlled by an OF controller over a unique OF channel using OpenFlow protocol. One OF controller can control multiple H-OFS instances using dedicated channels, or a dedicated OF controller can be deployed per switch. For each switch, up to two OF controllers can be deployed for redundancy. If two controllers are programmed, they can operate in either OFPCR_ROLE_EQUAL roles or in OFPCR_ROLE_MASTER and OFPCR_ROLE_SLAVE roles.

*Figure 31: SR OS/switch OF controller/switch architecture overview*




### 5.1.2 GRT-only and multiservice H-OFS modes of operations

SR OS supports two modes of operation for an H-OFS instance:

- GRT-only
- multiservice

The modes of operation are user-controlled per H-OFS instance by enabling or disabling the **switch-defined-cookie** option. Use the command in the following context to enable the **switch-defined-cookie** option:

- **MD-CLI**

```
configure openflow of-switch flowtable
```

- **classic CLI**

```
configure open-flow of-switch flowtable
```

For backward compatibility, GRT-only mode of operation is default but, because multiservice mode is a functional superset, Nokia recommends operating in multiservice mode whenever possible. The user can change the mode in which an H-OFS instance operates; however, the H-OFS instance must be administratively disabled first. This purges all the rules forcing the OF controller to reprogram the switch instance after it is re-enabled in a new mode. SR OS supports both H-OFS modes of operation concurrently for different switch instances.

Multiservice modes of operation uses part of the FlowTable cookie field (higher-order 32 bits) to provide the enhanced functionality; the lower-order FlowTable cookie bits are fully controlled by the OF controller. Table

9: Multiservice mode — higher-order bit flow table cookie encoding depicts higher-order bit Flow Table cookie encoding used when operating in the multiservice mode of operation.

*Table 9: Multiservice mode — higher-order bit flow table cookie encoding*

| sros-cookie Name | sros-cookie Type (bits 63...60) | sros-cookie Value (bits 59...32) | FlowTable Entry Interpretation Based on the sros-cookie |
|---|---|---|---|
| grt | 0000 | 0 | FlowTable rule is applicable to GRT instance (IES and router interfaces) |
| system | 1000 | 0 | FlowTable rule is applicable to system filters |
| service | 1100 | service-id for existing VPLS or VPRN service | FlowTable rule is applicable to an existing VPRN or VPLS service specified by the sros-cookie value |

To enable multiservice mode of operation, a user must embed the OF switch in an ACL filter policy, and because multiservice H-OFS supports a mix of VPRN, VPLS, GRT, and system rules, an additional scope of embedding must be selected.

Use the commands in the following contexts to embed the OF switch in an ACL filter policy. (GRT scope is used by default.)

- **MD-CLI**

  In the MD-CLI, use the **embed openflow grt**, **embed openflow vpls**, **embed openflow vprn**, or **embed openflow system** options in the following contexts.

  ```
  configure filter ip-filter
  configure filter ipv6-filter
  ```

- **classic CLI**

  In the classic CLI, use the **embed-filter open-flow service** or **embed-filter open-flow system** options in the following contexts.

  ```
  configure filter ip-filter
  configure filter ipv6-filter
  configure filter mac-filter
  ```

After embedding H-OFS instance, an ACL policy contains rules specific to a VPRN or VPLS service instance or to a GRT or to a system filter policy. Therefore, the ACL filter policy can only be used in the scope defined by H-OFS embedding.

Rules programmed by an OF controller with GRT, system, and service cookies specified are accepted even if the H-OFS instance is not embedded by a filter activated in a specific context. Rules programmed by an OF controller with a service cookie specified, when the service ID is not one of the supported service types, or when the service with the specified ID does not exist, are rejected with an error returned back to the controller. If an H-OFS is embedded into a line card policy with a specific service context, the embedding must be removed before that service is deleted.

The following table summarizes the main differences between the two modes of operation.

*Table 10: Differences between GRT mode and multiservice mode*

| Function | GRT Mode (no switch-defined-cookie) | Multiservice Mode (switch-defined-cookie) |
|---|---|---|
| Support OF on IES access interfaces | Yes | Yes |
| Support OF on router interfaces in GRT instance | Yes | Yes |
| Support OF on VPRN access and network interfaces | No (lack of native OF service virtualization) | Yes |
| Support OF on VPLS access and network interfaces | No (lack of native OF service virtualization) | Yes |
| Support port and VLAN in flowtable match (see the following section) | No | Yes |
| Support OF control of System ACL policies | No | Yes |
| Traffic steering actions | Forward, drop, redirect to LSP, Layer 3 PBR actions only | All |
| Scale | Up to ingress ACL filter policy entry scale | Up to OF system scale limit per H-OFS instance, and up to 64 534 entries per unique sros-cookie value |

### Restrictions

- See the SR OS R24.*x*.R*x* Software Release Notes for a full list of GRT, IES, VPRN, and VPLS interfaces that support OF control for multiservice mode.
- The 7450 ESS, 7750 SR, 7950 XRS, and VSR H-OFS always requires an sros-cookie to be provided for FlowTable operations and fails any operation without the cookie when the **switch-defined-cookie** command is enabled.
- OF no-match-action is not programmed in hardware for system filters, because system filters are chained to other filter policies and no-match-action would break the chaining.
- An H-OFS instance does not support overlapping of priorities (flow_priority value) within a single sros-cookie (type plus value). The supported values for priority differ based on a value for **switch-defined-cookie:**
  - H-OFS with the **switch-defined-cookie** command disabled
    - Valid flow_priority_range 1 to max-size – 1
    - flow_priority_value 0 is reserved (no match action)
  - H-OFS with the **switch-defined-cookie** command enabled
    - Valid flow_priority_range 1 to 65534
    - flow_priority_value 0 is reserved (no match action)

- flow_priority must map to a valid filter ID. The following items show how flow_priority is mapped to a filter policy entry ID:
  - H-OFS with the **switch-defined-cookie** command disabled

    filter entry ID = max-size – flow_priority + embedding offset
  - H-OFS with the **switch-defined-cookie** command enabled

    filter entry ID = 65535 – flow_priority + embedding offset
- When multiple H-OFS instances are embedded into a single ACL filter, no two H-OFS instances can program the same filter entry ID.

#### 5.1.2.1 Port and VLAN ID match in flow table entries

When operating in multiservice mode, SR OS H-OFS supports matching on port and VLAN IDs as part of Flow Table match criteria. When an OF controller specifies incoming port and VLAN values other than "any", the H-OFS instance translates them to an SR OS VPLS SAP (sros-cookie must be set to a valid VPLS service ID). If the translation does not result in an existing VPLS SAP, the rule is rejected and an error is returned to the controller.

A flow table rule with a port and VLAN ID match is programmed only if the matching SAP has this H-OFS instance embedded in its ACL ingress filter policy. Use the following commands to configure the SAP scope of embedding:

- **MD-CLI**

```
configure filter ip-filter embed openflow sap
configure filter ipv6-filter openflow embed openflow sap
```

- **classic CLI**

```
configure filter ip-filter embed-filter open-flow sap
configure filter ipv6-filter open-flow embed-filter sap
configure filter mac-filter open-flow embed-filter sap
```

See SR OS H-OFS port and VLAN encoding for required encoding of port and VLAN IDs.

The SR OS H-OFS supports a mix of rules with service scope and with SAP scope. For VPLS SAPs, an H-OFS instance must be embedded twice: after for the VPLS service and after for the SAP if both service-level and SAP-level rules are to be activated.

The following example shows the activation of service-level and SAP-level rules inside a single ACL policy.

**Example: MD-CLI**

```
[ex:/configure filter ip-filter "1"]
A:admin@node-2# info
    scope exclusive
    embed {
        openflow "ofs1" offset 100 {
            vpls "vpls100"
        }
        openflow "ofs1" offset 200 {
            vpls "vpls100"
            sap 1/1/2:2
        }
    }
```

### Example: classic CLI

```
A:node-2>config>filter>ip-filter# info
----------------------------------------------
            scope exclusive
            embed-filter open-flow "ofs1" service 5 offset 100
            embed-filter open-flow "ofs1" sap 1/1/2:2 offset 200
----------------------------------------------
```

### Restrictions

- Because an H-OFS instance does not support overlapping priorities within a single sros-cookie (type plus value), the priority for rules applicable to different SAPs within the same VPLS service must not overlap.
- Masking is not supported when adding a new flow table rule with a port and VLAN ID match.

## 5.1.3 Hybrid OpenFlow switch steering using filter policies

A router H-OFS instance is embedded into line card IPv4 and IPv6 filter policies to achieve OF-controlled Policy Based Routing (PBR). When H-OFS instance is created, embedded filters (IP and IPv6) required for that instance are automatically created. The filters are created with names, for example, "_tmnx_ofs_<ofs_name>", with the same name for IPv4 and IPv6 filters used.

If embedded filters cannot be allocated because of the lack of filter policy instances, the creation of an H-OFS instance fails. When the H-OFS instance is deleted, the corresponding embedded filters are freed.

The H-OFS can be embedded only in ingress filter policies on line cards/platforms supporting embedded filters and for services supporting H-OFS. Embedding of an H-OFS in filter policies on unsupported services is blocked. Embedding of an H-OFS in filter policies in unsupported direction or on unsupported hardware follows the general filter policy misconfiguration behavior and is not recommended. Unsupported match fields are ignored. Other match criteria may cause a packet to match an entry.

As soon as an H-OFS instance is created, the controller can program OF rules for that instance. For instance, the rules can be created before the H-OFS instance embedding into a filter policy or before a filter policy with H-OFS instance embedded being assigned to an interface. This allows the user to either pre-program H-OFS steering rules, or to disable the rules without removing them from a flow table by removing the embedding. An error is returned to the controller if it attempts to program rules not supported by the system. The following are examples of the errors returned:

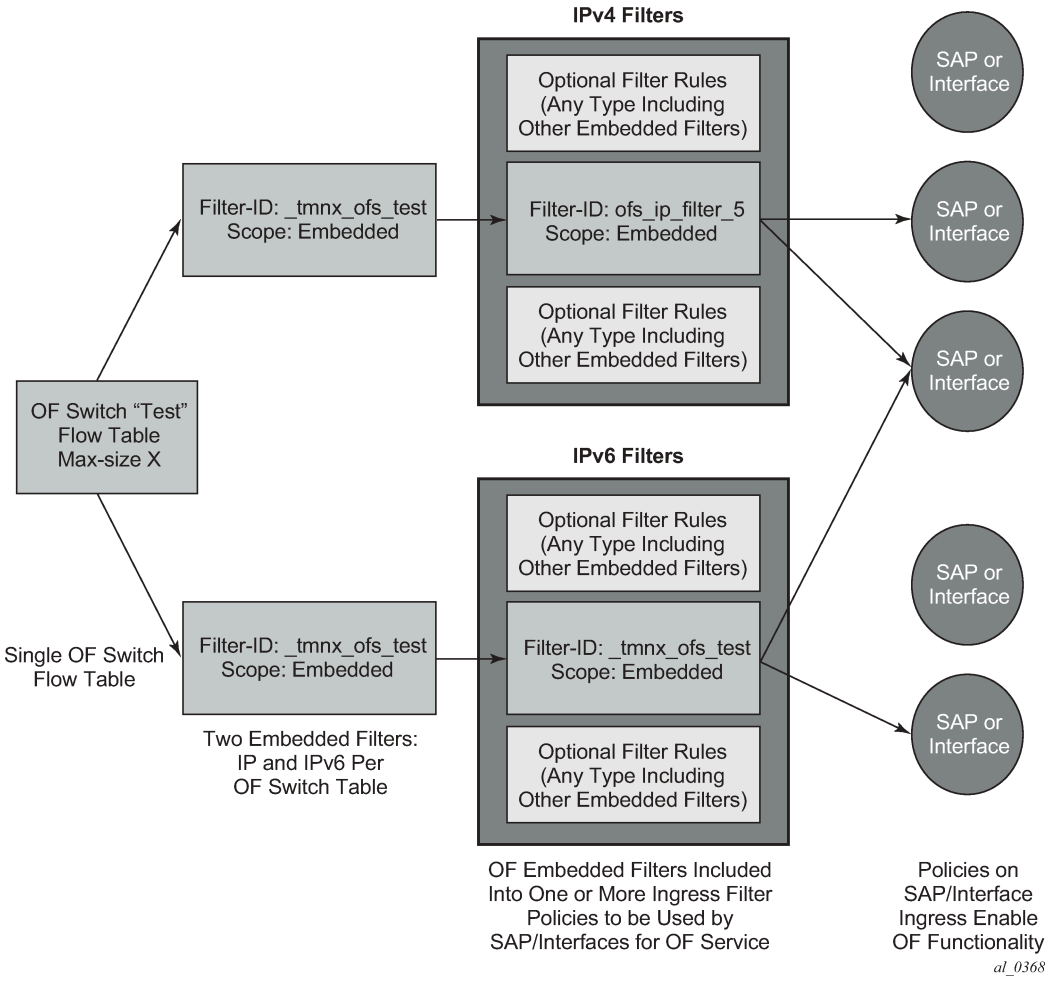
- unsupported instr: [OFPET_BAD_INSTRUCTION, OFPBIC_UNSUP_INST]
- unsupported action: [OFPET_BAD_ACTION, OFPBAC_BAD_TYPE]?
- unsupported output port: [OFPET_BAD_ACTION, OFPBAC_BAD_OUT_PORT]?
- unsupported match field: [OFPET_BAD_MATCH, OFPBMC_BAD_FIELD]?
- unsupported match value: [OFPET_BAD_MATCH, OFPBMC_BAD_VALUE]?
- output port invalid/deleted after flow_mod is sent to filter: OFPET_BAD_ACTION, OFPBAC_BAD_OUT_PORT]?

When the OF controller updates traffic steering rules, the Hybrid OpenFlow Switch updates the flow table rules. This automatically triggers programming of the embedded filter, which consequently causes instantiation of the rules for all services or interfaces that have a filter policy embedding this H-OFS instance. Embedded filter policy configuration/operational rules apply also to embedded filters auto-created for an H-OFS instance. MPLS cannot be deleted if OFS rules are created that redirect to an LSP.

The auto-created embedded filters can be viewed through CLI but cannot be modified or deleted through filter policy CLI/SNMP. The user can see the above embedded filters under show filter context, including the details about the filters, entries programmed, interface association, statistics, and so on.

For an H-OFS with the **switch-defined-cookie** command enabled, embedded filters are created for each unique context in the H-OFS instead.

*Figure 32: OF flow table mapping to router or switch service infrastructure example — switch-defined-cookie disabled*




The router allows mixing H-OFS rules from one or more H-OFS instances in a single filter policy. Co-existence of H-OFS rules in a single policy with CLI or SNMP programmed rules or BGP FlowSpec programmed rules in a single line card filter policy is also supported. When a management interface and an OF controller flow entry have the same filter policy entry, the management interface-created entry overrides the OF controller-created entry; see the embedded filter functional description. For mixing of the rules from multiple management entities, the controller should not program an entry in its Flow Table that would match all traffic, because this would stop evaluation of the filter policy.

The router supports HA for the OF Flow Table content and statistics. On an activity switch, the channel goes down and is reestablished by the newly active CPM. "Fail secure mode" operation takes place during channel reestablishment (OpenFlow rules continue to be applied to the arriving traffic). The OF controller

is expected to resynchronize the OF table when the channel is reestablished. On a router reboot or H-OFS instance shutdown, H-OFS Flow Table rules and statistics are purged. An H-OFS instance cannot be deleted unless the H-OFS instance is first removed from all embedding filter policies.

### 5.1.4 Hybrid OpenFlow switch statistics

The SR OS Hybrid OpenFlow switch supports statistics retrieval using the OpenFlow protocol. There are two types of statistics that can be collected:

1. Statistics for SR OS H-OFS logical ports

   Logical port statistics are available for RSVP-TE and MPLS-TP LSP logical ports. The non-zero statistics are returned as long as an LSP has statistics enabled through an MPLS configuration.

   Zero is always returned for logical port statistics for SR-TE LSPs when LSP statistics are not supported on SR-TE LSPs. The statistics can be retrieved regardless of whether an OF switch uses the specified LSP. The returned packet or byte values are an aggregate of all packets or bytes forwarded over the LSP.

   Statistics are not available for any other logical ports encodings.

2. Statistics for SR OS H-OFS flow table

   Flow table statistics can be retrieved for one or more flow table entries of an H-OFS. The returned packet/bytes values are based on ACL statistics collected in the hardware. An OpenFlow controller can retrieve statistics either directly from hardware or from the ACL CPM-based bulk request cache. The ACL cache is used when processing an OpenFlow statistics multipart aggregate request message (OFPMP_AGGREGATE), or when an OpenFlow statistics multipart flow message request (OFPM_FLOW) is translated to multiple flow table entries (a bulk request). When an OpenFlow multipart flow statistics request message (OFPM_FLOW) is translated to a single flow table entries request (a single entry request), the counters are read from hardware in real time.

   A combination of the two methods can be used to retrieve some flow table statistics from hardware in real time while retrieving other statistics from the cache. See Filter policy statistics for more information about ACL cache and ACL statistics.

   When the auxiliary channel is enabled, the switch sets up a dedicated auxiliary channel for statistics. See OpenFlow switch auxiliary channels for more information.

#### Operational notes

Consider the following operational notes:

- Flow Table statistics displayed through the CLI debugging tools are read in real time from hardware. However, to protect the system, executing CLI debugging tool commands within 5 s returns the same statistics for any flow that had its statistics read from hardware within the last 5 s. Use the following command to display flow table statistics.

  ```
  tools dump open-flow of-switch
  ```

- When retrieving FlowTable statistics at scale, Nokia recommends to either use bulk requests, or to pace single entry requests to obtain the balance between stats real-time accuracy and CPM activity.

### 5.1.5 OpenFlow switch auxiliary channels

The H-OFS supports auxiliary channels, as defined in OpenFlow version 1.3.1. The packet-in and statistics functions are supported on the auxiliary channels as well as on the main channel.

When the auxiliary channel is enabled on a switch (using the **aux-channel-enable** command), the switch sets up a dedicated auxiliary channel for statistics (Auxiliary ID 1) and a dedicated auxiliary channel for packet-in (Auxiliary ID 2) if a packet-in action is configured, to every controller for a specific H-OFS switch instance. Use the following command to enable the auxiliary channel:

- **MD-CLI**

```
configure openflow of-switch aux-channel true
```

- **classic CLI**

```
configure open-flow of-switch aux-channel-enable
```

Auxiliary connections use the same transport as the main connection. The switch handles any requests over any established channel and respond on the same channel even if a specific requested auxiliary channel is available.

The H-OFS instance uses the packet-in connection for packet-in functionality by default and expects (but does not require) the controller to use the statistics channel for statistics processing by default.

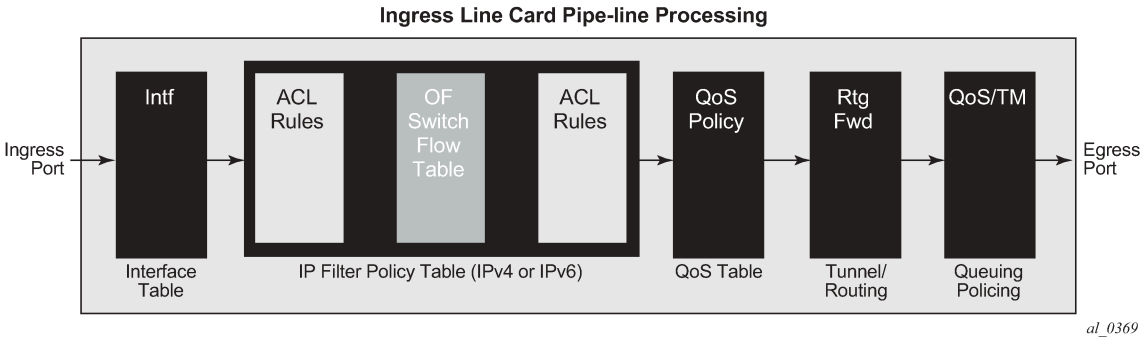
The switch uses the auxiliary channels (packet-in for packet-in-specific requests and statistics for statistics-specific requests) as long as they are available. If they are not available, the switch uses the next available auxiliary channel. If none of the auxiliary channels are available, the main channel is used.

Auxiliary connections can be enabled or disabled without shutting down the switch.

### 5.1.6 Hybrid OpenFlow switch traffic steering details

As described in OpenFlow switch auxiliary channels, an update to an OpenFlow Switch's flow table results in the embedded filter updates, which triggers an update to all filter policies embedding those filters. The router automatically downloads the new set of rules to the line cards as defined through service configuration. The following figure shows how the rules become part of an ingress line card pipeline.

*Figure 33: OpenFlow switch embedding in ingress pipeline processing*



### 5.1.6.1 SR OS H-OFS logical port

Logical ports are used in OpenFlow to encode switch-specific ports. SR OS H-OFS uses logical ports in steering actions by encoding PBR targets. Table 11: Encoding and supported logical port types lists logical port types supported by SR OS H-OFS:

*Table 11: Encoding and supported logical port types*

| Bits 31..28 | Bits 27..24 | Bits 24..0 |
|---|---|---|
| Logical port type (LPT) | Logical port type sub-type (LPT-S) | Logical port type value (LPT-V) — always padded with leading zeros |

The following encoding shows logical port types supported by SR OS H-OFS.

```
RSVP LSP: LPT: 0100, LPT-S: 0000 (tunnel), LPT-V: RSVP TE Tunnel ID
MPLS-TP LSP: LPT: 0100, LPT-S: 0000 (tunnel), LTP-V: MPLS-TP Tunnel Number
SR-TE LSP: LPT: 0100, LPT-S: 0000 (tunnel), LTP-V: SR-TE LSP Index
GRT instance: LPT: 0100, LPT-S: 0001 (L3 routing instance), LPT-V: 0
VPRN Id: LPT: 0100, LPT-S: 0001 (L3 routing instance), LPT-V: VPRN Service ID for a
VPRN instance configured on the system, NAT: LPT 0100, LPT-S: 0020 (NAT), LPT-V: 0
```

OF is limited to a 24-bit service ID value range (a subset of VPRN IDs supported by the SR OS system).

Logical port values other than RSVP-TE LSP, SR-TE LSP, and MPLS-TP LSP require H-OFS with the **switch-defined-cookie** command enabled. Only tunnel-encoded ports are stored in the H-OFS logical port table. Therefore, functionality such as retrieving statistics per port is not available for logical ports that are not stored in the H-OFS logical port table.

### 5.1.6.2 SR OS H-OFS port and VLAN encoding

The OF controller can use port and VLAN values other than "ANY" for VPLS SAP match and for VPLS steering to SAP for H-OFS instances with the **switch-defined-cookie** command enabled.

To specify a port in an OF message, SR OS TmnxPortId encoding must be used. The allowed values are those for Ethernet physical ports and LAG.

The following table shows how OXM_OF_VLAN_ID and experimenter OFL_OUT_VLAN_ID fields are used to encode VLAN tags.

*Table 12: VLAN tag encoding*

| NULL tag, dot1Q tag, inner QinQ tag VlanId | Outer QinQ tag VlanId |
|---|---|
| OXM_OF_VLAN_VID | OFL_OUT_VLAN_ID (Experimenter field uses same encoding as OXM_OF_VLAN_VID) |

The following table shows how OF programmed values are translated to SR OS SAPs.

*Table 13: Translation of OF programmed values to SR OS SAPs*

| OXM_OF_ IN_PORT | OXM_OF_ VLAN_VID | OFL_ OUT_ VLAN_ID | Matching SAP SR OS Encoding | Supported in flow_add | Supported in flow_ mod flow_ del mp_ req | Comment |
|---|---|---|---|---|---|---|
| TmnxPort Id for port or LAG | Value: 0x0000<br>Mask: Absent | Must be absent | port-id<br>lag-id | ✓ | ✓ | Mask must be absent |
| TmnxPort Id for port or LAG | Value: 0x1yyy, yyy encodes qtag1<br>Mask: Absent | Must be absent | port-id:qtag1<br>lag-id:qtag1 | ✓ | ✓ | Mask must be absent |
| TmnxPort Id for port or LAG | Value: 0x1FFF<br>Mask: Absent | Must be absent | port-id:*<br>lag-id:* | ✓ | ✓ | Mask must be absent |
| TmnxPort Id for port or LAG | Value: 0x1000<br>Mask: 0x1000 | Must be absent | port-id: any<br>lag-id: any<br>where "any" is either * or a valid VLAN-ID (but not NULL) | | ✓ | Mask must be 0x1000 |
| TmnxPort Id for port or LAG | Value: 0x1yyy, yyy encodes qtag2<br>Mask: Absent | Value: 0x1zzz,<br>zzz encodes qtag1<br>Mask: Absent | port-id:qtag1.qtag2<br>lag-id:qtag1.qtag2 | ✓ | ✓ | Mask must be absent |
| TmnxPort Id for port or LAG | Value: 0x1FFF<br>Mask: Absent | Value: 0x1zzz,<br>zzz encodes qtag1<br>Mask: Absent | port-id: qtag1.*<br>lag-id: qtag1.* | ✓ | ✓ | Mask must be absent |
| TmnxPort Id for port or LAG | Value: 0x1FFF<br>Mask: Absent | Value: 0x1FFF | port-id: *.*<br>lag-id: *.* | ✓ | ✓ | Mask must be absent |

| OXM_OF_ IN_PORT | OXM_OF_ VLAN_VID | OFL_ OUT_ VLAN_ID | Matching SAP SR OS Encoding | Supported in flow_add | Supported in flow_ mod flow_ del mp_ req | Comment |
|---|---|---|---|---|---|---|
| | | Mask: Absent | | | | |
| TmnxPort Id for port or LAG | Value: 0x1000<br>Mask: 0x1000 | Value: 0x1zzz,<br>zzz encodes qtag1<br>Mask: Absent | port-id: qtag1.any<br>lag-id: qtag1.any<br>where any is either * or a valid VLAN-ID (but not NULL) | | √ | Mask must be absent for OFL_ OUT_ VLAN_VID |
| TmnxPort Id for port or LAG | Value: 0x1000<br>Mask: 0x1000 | Value: 0x1FFF<br>Mask: Absent | port-id: *.any<br>lag-id: *.any<br>where "any" is either * or a valid VLAN-ID (but not NULL) | | ✓ | Mask must be absent for OFL_ OUT_ VLAN_VID |
| TmnxPort Id for port or LAG | Value: 0x1000<br>Mask: 0x1000 | Value: 0x1000<br>Mask: 0x1000 | port-id: any.any<br>lag-id: any.any<br>where "any" is either * or a valid VLAN-ID (but not NULL) | | ✓ | Masks must be 0x1000 |
| TmnxPort Id for port or LAG | Value: 0x0000<br>Mask: Absent | Value: 0x1FFF<br>Mask: Absent | port-id: *.null | ✓ | ✓ | Mask must be absent |

### 5.1.6.3 Redirect to IP next-hop

A router supports redirection of IPv4 or IPv6 next-hop for traffic arriving on a Layer 3 interface. An OF controller can rely on this functionality and program PBR next-hop steering actions for H-OFS instances with the **switch-defined-cookie** command enabled using the following OF encoding.

```
ALU_IPD_EXPERIMENTER_ID: 0x000025BA
ALU_AXN_REDIRECT_TO_NEXTHOP: 2
```

```
flow_mod:
instruction= OFPIT_WRITE_ACTION/OFPIT_APPLY_ACTION,
action= OFPAT_EXPERIMENTER(ALU_AXN_REDIRECT_TO_NEXTHOP),
encoding:
struct alu_axn_redirect_to_nhopv4{
uint16_t type;                   /* OFPAT_EXPERIMENTER. */
uint16_t len;                    /* Total Length is a multiple of 8. */
uint32_t experimenter;           /* Experimenter ID vendor unique*/
uint8_t    redirect_type;        /* Type = 1 for Nhop*/
uint8_t    flags;                /* flags is 0-7 bits:
     Bit 0 = Ipv4,
     Bit 1 = Ipv6,
     Bit 2   = indirect
     */
uint8_t     pad[2];
uint32_t   ipaddr;                /* ipv4 addr */
unit8_t     pad[0];               /* Not needed */
}; ASSERT(sizeof(alu_axn_redirect_to_nhopv4) == 16)
struct alu_axn_redirect_to_nhopv6{
uint16_t type;                   /* OFPAT_EXPERIMENTER. */
uint16_t len;                    /* Total Length is a multiple of 8. */
uint32_t experimenter;           /* Experimenter ID vendor unique*/
uint8_t    redirect_type;        /*  Type = 1 for Nhop*/
uint8_t    flags;                /* flags is 0-7 bits:
     Bit 0 = Ipv4,
     Bit 1 = Ipv6,
     Bit 2   = indirect
     */
uint8_t       pad[2];
uint128_t   ip6addr;              /* ipv6 addr */
unit8_t     pad[4];               /* Make total len multiple of 8 */
}; ASSERT(sizeof(alu_axn_redirect_to_nhopv6) == 32)
```

In case of erroneous programming, the following experimenter-specific errors are returned to the controller.

```
enum alu_err_exp_class{
ALU_ERR_CLASS_RD_TO_SDP       = 0,
ALU_ERR_CLASS_RD_TO_NHOP      = 1,
}
enum alu_err_subtype_redirect_to_nhop
{
ALU_ERR_RN_INVALID_FLAGS      = 0
ALU_ERR_RN_INVALID_ARGS       = 1
ALU_ERR_RN_INVALID_ADDR       = 2
}
```

### 5.1.6.4 Redirect to GRT instance or VRF instance

A router supports redirection of IPv4 or IPv6 traffic arriving on a Layer 3 interface to a different routing instance (GRT or VRF). An OF controller can rely on this functionality and program PBR actions for GRT/VRF steering for H-OFS instances with the **switch-defined-cookie** command enabled using the following OF encoding.

```
flow_mod:
instruction type: OFPIT_WRITE_ACTIONS/OFPIT_APPLY_ACTION,
action type: OFPAT_OUTPUT,
```

port= SR OS LOGICAL port encoding GRT or VPRN Service ID as described in the SR OS H-OFS logical port section.

Because a 24-bit value is used to encode the VPRN service ID in the logical port, redirection to a VPRN service with a service ID above that range is not supported.

### 5.1.6.5 Redirect to next-hop and VRF/GRT instance

A router supports redirection of IPv4 or IPv6 traffic arriving on a Layer 3 interface to a different routing instance (GRT or VRF) and next-hop IP at the same time. An OF controller can rely on this functionality and program PBR steering action for H-OFS instances with the **switch-defined-cookie** command enabled using the following OF encoding.

```
ALU_IPD_EXPERIMENT_ID:0X000025BA
ALU_AXN_REDIRECT_TO_NEXTHOP:2
flow_mod:
Instruction 1:
instruction=OFPIT_WRITE_ACTION/OFPIT_APPLY_ACTION
action=OFPAT_EXPERIMENTER(ALU_AXN_REDIRECT_TO_NEXTHOP),
```

Encoding as described in the Redirect to IP next-hop section (indirect flag must be set).

```
Instruction 2:
instruction type: OFPIT_WRITE_ACTIONS/OFPIT_APPLY_ACTION,
action type: OFPAT_OUTPUT,
```

port= SR OS LOGICAL port encoding GRT or VPRN Service ID as described in the SR OS H-OFS logical port section.

### 5.1.6.6 Redirect to ESI (Layer 2)

The router supports redirection of IPv4 or IPv6 traffic arriving on a Layer 2 interface to an Ethernet Segment Identifier (ESI) with an EVPN control plane. An OF controller can program Layer 2 ESI steering with the **switch-defined-cookie** command enabled using the following OF encoding.

```
flow_mod:
    instruction type: OFPIT_WRITE_ACTIONS/OFPIT_APPLY_ACTION,
    action type: OFPAT_EXPERIMENTER(ALU_AXN_REDIRECT_TO_ESI_L2)
    encoding:

struct alu_axn_redirect_to_ESI_L2{
    uint16_t  type;              /* OFPAT_EXPERIMENTER. */
    uint16_t  len;               /* Total  Length is a multiple of 8. */
    uint32_t  experimenter;      /* Experimenter ID vendor unique*/
    uint8_t   redirect_type ;    /*  Type = 3 for ESI*/
    uint8_t   flags;             /*  flags is 0-7 bits:
                                     Value   0 = L2,
                                 */
    uint8_t    esi[10];          /* 10 byte ESI */
    uint32_t   svcId;            /* Svc-Name Using the OF Encoding */
}; ASSERT(sizeof(alu_axn_redirect_to_ESI_L2) == 24)
```

### 5.1.6.7 Redirect to ESI (Layer 3)

The router supports redirection of IPv4 or IPv6 traffic arriving on a Layer 3 interface to an ESI with an EVPN control plane. An OF controller can program Layer 3 ESI steering with the **switch-defined-cookie** command enabled using the following OF encoding.

```
flow_mod:
    instruction type: OFPIT_WRITE_ACTIONS/OFPIT_APPLY_ACTION,
    action type: OFPAT_EXPERIMENTER(ALU_AXN_REDIRECT_TO_ESI_L3)
    encoding:

struct alu_axn_redirect_to_ESI_L3_V4{
    uint16_t  type;               /* OFPAT_EXPERIMENTER. */
    uint16_t  len;                /* Total  Length is a multiple of 8. */
    uint32_t  experimenter;       /* Experimenter ID vendor unique*/
    uint8_t   redirect_type ;     /*  Type = 3 for ESI*/
    uint8_t   flags;              /*  flags is 0-7 bits:
                                      Value   1 = L3 (ipv4)
                                  */
    uint8_t    esi[10];           /* 10 byte ESI */
    uint32_t   svcId;             /* Svc-Name Using the OF Encoding */
    uint32_t   sf-ip;             /* v4 address of sf-ip */
    uint32_t   ifIndex;           /* interface id*/
}; ASSERT(sizeof(alu_axn_redirect_to_ESI_L3_V42) == 32)

struct alu_axn_redirect_to_ESI_L3_V6{
    uint16_t  type;               /* OFPAT_EXPERIMENTER. */
    uint16_t  len;                /* Total  Length is a multiple of 8. */
    uint32_t  experimenter;       /* Experimenter ID vendor unique*/
    uint8_t   redirect_type ;     /*  Type = 1 for Nhop*/
    uint8_t   flags;              /*  flags is 0-7 bits:
                                      Value = 2 = L3 (ipv6)
                                  */
    uint8_t    esi[10];           /* 10 byte ESI */
    uint32_t   svcId;             /* Svc-Name Using the OF Encoding */
    uint128_t  sf-ip;             /* v6 address of sf-ip */
    uint32_t   ifIndex;           /* interface id*/
    uint8_t    pad[4];
}; ASSERT(sizeof(alu_axn_redirect_to_ESI_L3_V6) == 48)
```

### 5.1.6.8 Redirect to ESI IP VAS-interface router

The router supports redirection of IPv4 or IPv6 traffic arriving on a Layer 3 interface to a VAS interface bound to an ESI with an EVPN control plane. In this encoding, the SF-IP address represents the VAS interface address, and the ifIndex is the VAS interface ID. An OF controller can program Layer 3 steering with the **switch-defined-cookie** command enabled using the following OF encoding.

```
flow_mod:
    instruction type: OFPIT_WRITE_ACTIONS/OFPIT_APPLY_ACTION,
    action type: OFPAT_EXPERIMENTER(ALU_AXN_REDIRECT_TO_ESI_L3)
    encoding:

struct alu_axn_redirect_to_ESI_L3_V4{
    uint16_t  type;               /* OFPAT_EXPERIMENTER. */
    uint16_t  len;                /* Total Length is a multiple of 8. */
    uint32_t  experimenter;       /* Experimenter ID vendor unique*/
    uint8_t   redirect_type ;     /*  Type = 2 for ESI*/
    uint8_t   flags;              /*  flags is 0-7 bits:
```

```
                                    Value 2 = L3 (ipv4)
                                */
    uint8_t    esi[10];
    uint32_t   svcId;            /* Svc-Name Using the OF Encoding */
    uint32_t   vas-ip;           /* v4 address of sf-ip */
    uint32_t   ifIndex;          /* vas interface id*/
}; ASSERT(sizeof(alu_axn_redirect_to_ESI_L3_V4) == 24)

struct alu_axn_redirect_to_ESI_L3_V6{
    uint16_t  type;              /* OFPAT_EXPERIMENTER. */
    uint16_t  len;               /* Total Length is a multiple of 8. */
    uint32_t  experimenter;      /* Experimenter ID vendor unique*/
    uint8_t   redirect_type ;    /*  Type = 2 for ESI*/
    uint8_t   flags;             /*  flags is 0-7 bits:
                                    Value 4 = L3 (ipv6)
                                */
    uint8_t    esi[10];          /* 10 byte ESI */
    uint32_t   svcId;            /* Svc-Name Using the OF Encoding */
    uint128_t  vas-ip;           /* v6 address of sf-ip */
    uint32_t   ifIndex;          /* vas interface id*/
    uint8_t    pad[4]
}; ASSERT(sizeof(alu_axn_redirect_to_ESI_L3_V6) == 40)
```

### 5.1.6.9 Redirect to LSP

The router supports traffic steering to an RSVP, MPLS-TP, IPv4 SR-TE, or IPv6 SR-TE LSP. The following shows the OF encoding to be used by an OF controller.

```
flow_mod:
instruction type: OFPIT_WRITE_ACTIONS or OFPIT_APPLY_ACTION,
action type: OFPAT_OUTPUT,
```

The port uses SR OS LOGICAL port encoding RSVP-TE, SR-TE, or MPLS-TP LSP as described in the SR OS H-OFS logical port section.

An LSP received in a flow rule is compared against those in the H-OFS logical port table. If the table does not contain the LSP, the rule programming fails. Otherwise, the rule is installed in an ACL filter. As long as any path within the LSP is UP, the redirect rule forwards unicast IPv4 or IPv6 traffic on the current best LSP path by adding an LSP transport label and, in the case of IPv6 traffic, also adding an explicit NULL label.

When an LSP in the H-OFS logical port table goes down, the OF switch removes the LSP from its logical port table and notifies the controller of that fact if the logical port status reporting is enabled. It is up to the OF controller to decide whether to remove rules using this LSP. If the rules are left in the flow table, the traffic that was to be redirected to this LSP instead is subject to a forward action for this flow rule. If the controller does not remove the entries and the system reuses the LSP identified for another LSP, the rules left in the flow table start redirecting traffic onto this new LSP.

In some deployments, an SDN controller may need to learn from the router H-OFS logical ports status. To support this function, the OF switch supports optional status reporting using asynchronous OF protocol messages for ports status change.

### 5.1.6.10 Redirect to NAT

The router supports redirection of IPv4 traffic arriving on a Layer 3 interface for ISA NAT processing. An OF controller can program NAT steering for H-OFS instances with the **switch-defined-cookie** command enabled using the following OF encoding.

```
flow_mod:
    instruction type: OFPIT_WRITE_ACTIONS/OFPIT_APPLY_ACTION,
    action type: OFPAT_OUTPUT,
```

The port uses SR OS LOGICAL port encoding as described in the SR OS H-OFS logical port section.

### 5.1.6.11 Redirect to SAP

For traffic arriving on a VPLS interface, a router supports PBF to steer traffic over another VPLS SAP in the same service. An OF controller can rely on this functionality and program PBF steering action for H-OFS instances with the **switch-defined-cookie** command enabled using the following OF encoding.

```
flow_mod:
instruction type: OFPIT_WRITE_ACTIONS or OFPIT_APPLY_ACTION,
Action 1:
action type: OFPAT_OUTPUT,
```

The port uses encoding as described in the SR OS H-OFS port and VLAN encoding section.

```
Action 2:
action type=OFPAT_SET_FIELD
```

OXM TLVs encode SAP VLANs as described in the SR OS H-OFS port and VLAN encoding section:

```
- OXM_OF_VLAN_VID
- OFL_OUT_VLAN_ID (optional)
```

### 5.1.6.12 Redirect to SDP

For traffic arriving on a VPLS interface, a router supports PBF to steer traffic over a VPLS SDP in the same service. An OF controller can rely on this functionality and program PBF steering action for H-OFS instances with **switched-defined-cookie** enabled using the following OF encoding.

```
ALU_IPD_EXPERIMENTER_ID:  0x000025BA
ALU_AXN_REDIRECT_TO_SDP: 1
flow_mod:
instruction= OFPIT_WRITE_ACTIONS/OFPIT_APPLY_ACTIONS,
action= OFPAT_EXPERIMENTER(ALU_AXN_REDIRECT_TO_SDP),
encoding:
struct alu_axn_redirect_to_sdp{
uint16_t type;                   /* OFPAT_EXPERIMENTER. */
uint16_t len;                    /* Total Length is a multiple of 8. */
uint32_t experimenter;           /* Experimenter ID vendor unique*/
uint8_t    redirect_type;        /* Type = 0 for SDP*/
uint8_t    flags;               /
* Flags that can be used to denote    info(reserved)*/
uint16_t sdp-id;                 /* Sdp-id*/
```

```
uint32_t vcId;                   /* Vc-id*/
unit8_t    pad[0];                /* Not needed */
}; ASSERT(sizeof(alu_axn_redirect_to_sdp) == 16)
```

In case of erroneous programming, the following experimenter-specific errors are returned to the controller:

```
enum alu_err_exp_class
{
ALU_ERR_CLASS_RD_TO_SDP      = 0,
ALU_ERR_CLASS_RD_TO_NHOP     = 1,
}
enum alu_err_redirect_to_sdp
{
ALU_ERR_RS_INVALID_FLAGS     = 0
ALU_ERR_RS_INVALID_ARGS      = 1
ALU_ERR_RS_INVALID_SDP_ID    = 2
ALU_ERR_RS_INVALID_VC_ID     = 3
}
```

### 5.1.6.13 Redirect to a specific LSP used by a VPRN service

The router supports traffic steering within a VPRN, enabling the transport tunnels used by the SDP to be used for specific flows redirected from the system-selected default. This redirection enables large bandwidth flows to be moved to an alternative LSP.

For matching ingress traffic on a VPRN, the **switch-defined-cookie** command must be enabled, with the cookie encoded to match the ingress VPRN's service ID.

Traffic can be redirected to the following:

- the default PE and a different LSP
- a different PE and the default LSP
- the default PE and the default LSP (traffic that may otherwise egress a SAP takes a specified BGP next hop)
- the default PE and a different VRF
- a different PE, the default LSP, and a different prefix

Parameters must be matched in the OF encoding to steer traffic.

```
flow_mod:
instruction type: OFPIT_WRITE_ACTIONS/OFPIT_APPLY_ACTION,
```

```
Action 1:
action type: OFPAT_EXPERIMENTER
ALU_IPD_EXPERIMENTER_ID:  0x000025BA
ExpType= ALU_AXN_REDIRECT_TO_NEXTHOP,
```

```
Action 2:
action type: OFPAT_OUTPUT,
```

port= SR OS LOGICAL port encoding RSVP-TE, MPLS-TP LSP, or segment routing, as described in SR OS H-OFS logical port section.

Action 3 (optional): to redirect to a different VPRN

```
Action 3:
action type: OFPAT_EXPERIMENTER
ALU_IPD_EXPERIMENTER_ID:  0x000025BA
ExpType= ALU_AXN_REDIRECT_TO_VPRN,
```

Encoding:

```
struct alu_axn_redirect_to_vprn {
    uint16_t              type;                  /* OFPAT_EXPERIMENTER =>  ff ff */
    uint16_t              len;
    uint32_t              experimenter;           /
* Vendor specific experimenter id => 00 00 25 ba */
    uint8_t               exp_axn_type;          /*  type => 03 */
    uint8_t               exp_axn_flags;         /* flag => any value is accepted */
    uint8_t               pad[2];                /* pad => 00 00  */
    uint32_t              vprn;                  /* vrpn svc id */
};ASSERT(sizeof(alu_axn_redirect_to_vprn) == 16)
```

Action 4 (optional): to redirect to a different prefix

```
Action 4:
action type: OFPAT_SET_FIELD
```

Field is an IP destination address. Subnet masks are not supported in the set_field instruction.

### 5.1.6.14 Forward action

An OF controller can program forward action, when a specific flow is to be forwarded using regular router forwarding. This would be a default behavior if the filter policy embedding this OF switch instance has a default-action forward and no filter policy rule matches the flow. To implement forward action, the following OF encoding is used.

```
flow_mod:
instruction type: OFPIT_WRITE_ACTIONS or OFPIT_APPLY_ACTION,
action type: OFPAT_OUTPUT,
port= NORMAL
```

where NORMAL is an OF reserved value.

### 5.1.6.15 Drop action

An OF controller can program a drop action, when packets of a specific flow are to be dropped. To implement a drop action, the OF encoding is a wildcard rule with empty action-set.

### 5.1.6.16 Default no-match action

Packets that do not match any of the flow table entries programmed by the controller are subject to a default action. Use the following command to configure the default action:

- **MD-CLI**

```
configure openflow of-switch flowtable mismatch-action fall-through
```

- **classic CLI**

```
configure open-flow of-switch flowtable no-match-action fall-through
```

Three possible no-match actions are supported: drop, fall-through (packets are forwarded with regular processing by the router), and packet-in.

The packet-in action causes packets that do not match entries in the flow table, as programmed by the OpenFlow controller, to be extracted and sent to the controller in a flow-controlled manner. Because EQUAL is supported, packet-in messages are sent to all controllers in the UP state. To protect the controller, only the first packet of a specific 5-tuple flow (source IP address, destination IP address, source port, destination port, protocol) to which the no-match action is applied is sent to the controller. This 5-tuple flow context ages out after 10 s. Each switch instance maintains contexts for up to 8192 outstanding packet-in messages to the controller. If the packet-in action is used, an auxiliary channel should be enabled for packet-in processing. Use the following command to enable the packet-in processing:

- **MD-CLI**

```
configure openflow of-switch aux-channel true
```

- **classic CLI**

```
configure open-flow of-switch aux-channel-enable
```

A count of packets to which packet-in is applied is also available through the OpenFlow channel statistics.

### 5.1.6.17 Programming of DSCP remark action

The router supports DSCP remarking of IPv4 and IPv6 packets arriving on VPLS, VPRN, GRT, and system interfaces for OFS instances with the **switch-defined-cookie** command enabled using the following OF encoding.

```
flow_mod:
    instruction type: OFPIT_METER
    action type:  with the meterId.
```

The meters are configured using meter modification messages, and are configured before the flow messages are sent with meter instruction.

```
typedef struct tOfpMeterModMsg
{
    tOfpMsgHeader       msgHdr;
    uint16_t            mtrCommand;    /* One of OFP_MTR_CMD_*. */
    uint16_t            mtrConfig;     /* bitmap of OFP_MTR_CFG_*. */
    uint32_t            mtrId;         /* Meter instance. */
    tOfpMeterBandHeader bands[0];      /* The band list length is inferred from
                                          the length field in the msgHdr. */
} tOfpMeterModMsg;

typedef struct tOfpMeterBandHeader
{
    uint16_t            bandType;      /* One of OFP_MTR_BAND_*. */
```

```
    uint16_t            length;        /* Length in bytes of this band. */
    uint32_t            rate;          /* Rate for this band. */
    uint32_t            burstSize;     /* Size of bursts. */
} tOfpMeterBandHeader;

typedef enum eOfpMeterBandType
{
    OFP_MTR_BAND_DROP         = 1,             /* Drop packet. */
    OFP_MTR_BAND_DSCP_REMARK  = 2,             /* Remark DSCP in the IP header. */
    OFP_MTR_BAND_EXPERIMENTER = 0xFFFF         /* Experimenter meter band. */
} eOfpMeterBandType;

typedef struct tOfpMeterBandDscpRemark
{
    tOfpMeterBandHeader  bandHdr;      /* OFP_MTR_BAND_DSCP_REMARK */
    uint8_t              precLevel;    /* Number of drop precedence level to add */
    uint8_t              pad[3];
} tOfpMeterBandDscpRemark;
```

### 5.1.7 Support for secondary actions for PBR/PBF redundancy

The router supports "primary and secondary action" for filters (see Primary and secondary filter policy action for PBR/PBF redundancy). OpenFlow programming for multiple filter actions is also supported, as follows:

- **Layer 2 PBF redundancy**

  Two PBF actions (SAP and SDP) with optional sticky destination and pbr-on-down

- **Layer 3 PBR redundancy**

  Two PBR actions (IP next-hop and VRF) with optional sticky destination and pbr-on-down

- **Layer 3 PBR redundancy**

  Two PBR actions (next-hop and VRF) with optional sticky destination and pbr-on-down, with DSCP remarking as an extended action

The router supports multi-action using the OpenFlow version 1.3.1 Required Action: Group (For more details, see Section 6.4, Flow Table Modification Messages, Section 6.5, Group Table Modification Messages, and Section 5.6.1, Group Types with group type of fast failover of the TS-007, OpenFlow Switch Specification Version 1.3.1 (OpenFlow-hybrid switches)).

Redundancy uses fast failover group modeling as per the OpenFlow specification with two buckets, with liveliness detection provided by the filter module. Note that failover operates independently of the OpenFlow controller.

The router supports the programming of **pbr-on-down-override** and **sticky-dest** using an experimenter, as follows.

```
instruction type: OFPIT_WRITE_ACTIONS/OFPIT_APPLY_ACTION,
action type: OFPAT_EXPERIMENTER(ALU_AXN_PBFPBR_REDUNDANCY)
encoding:
struct alu_axn_PBFPBR_Redundancy{
   uint16_t  type;         /* OFPAT_EXPERIMENTER. */
   uint16_t  len;          /* Total  Length is a multiple of 8. */
   uint32_t  experimenter;  /* Experimenter ID vendor unique*/
   uint8_t   redirect_type ;  /*  Type = 4 for PBR/PBF redundancy*/
   uint8_t   flags;          /* flags is 0-7 bits:
                               Bit 0 for pbr-down-action-override
                               Bit 1 for sticky-dest
```

```
                              */
    uint32_t   portId;       /* to specify pbr-down action portId 0
                            for drop, OFP_PORT_ID_NORMAL for forward
                            OFP_PORT_ID_ANY for filter-default-action
                            */
    uint32_t   holdTime;    /* Value between 0-65535 seconds for
                             sticky dest */
};
```

## 5.2 Configuration notes

The following information describes OF implementation restrictions:

- SR OS Hybrid OpenFlow Switch requires a software upgrade only and can be enabled on any switch. For full functionality, performance, and future scale, CPM4 or newer control cards are recommended.
- Some platforms may not support all OF functionality based on the underlying hardware. For example, if the underlying hardware does not support IPv6, OF IPv6 functionality is not supported. If the underlying hardware does not support redirect to LSP, redirect action is ignored.
- Each flow in an OF flow table must have unique priority. Overlap is not supported.
- Timed expiry of the flow entries is not supported.
- The implementation is compliant by design with OpenFlow specification as applicable to supported router functionality only.

# 6 Cflowd

## 6.1 Cflowd overview

Cflowd is a tool used to obtain samples of IPv4, IPv6, MPLS, and Ethernet traffic data flows through a router. Cflowd enables traffic sampling and analysis by ISPs and network engineers to support capacity planning, trends analysis, and characterization of workloads in a network service provider environment.

Cflowd is also useful for traffic engineering, network planning and analysis, network monitoring, developing user profiles, data warehousing and mining, as well as security-related investigations. Collected information can be viewed several ways such as in port, AS, or network matrices, and pure flow structures. The amount of data stored depends on the cflowd configurations.

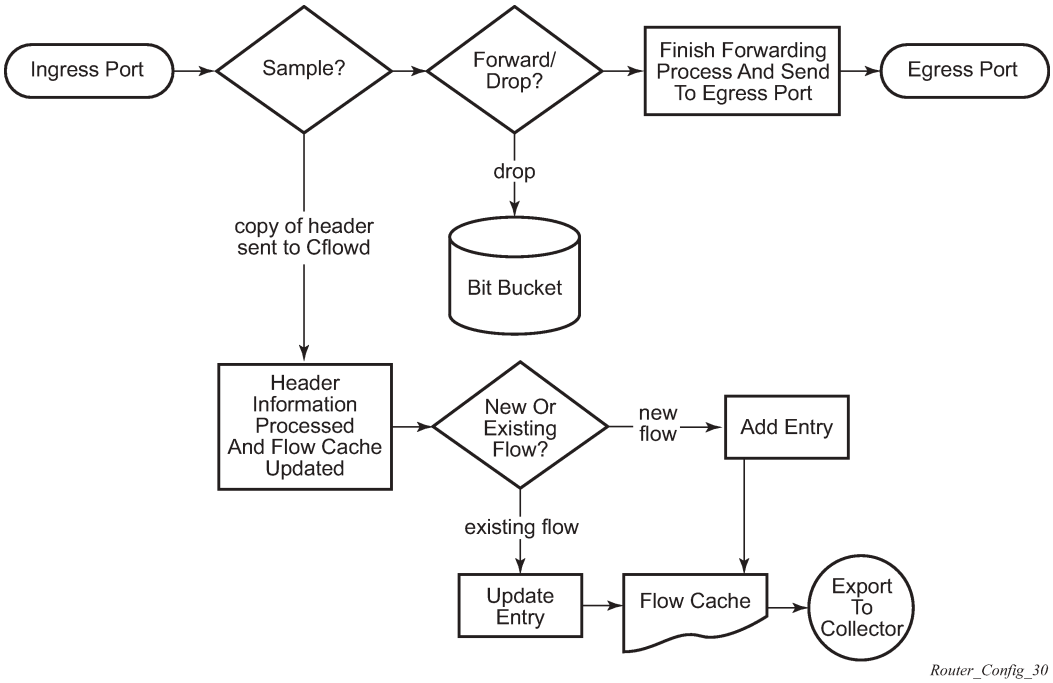
Cflowd maintains a list of data flows through a router. A flow is a unidirectional traffic stream defined by several characteristics such as source and destination IP addresses, source and destination ports, inbound interface, IP protocol and ToS bits.

When a router receives a packet for which it currently does not have a flow entry, a flow structure is initialized to maintain state information about that flow, such as the number of bytes exchanged, IP addresses, port numbers, AS numbers, and so on. Each subsequently sampled packet matching the same command options of the flow contributes to the byte and packet count of the flow until the flow is terminated and exported to a collector for storage.

### 6.1.1 Operation

Figure 34: Basic cflowd steps shows the basic operation of the cflowd feature. This sampled flow is only used to describe the basic steps that are performed. It is not intended to specify implementation.

*Figure 34: Basic cflowd steps*




1. As a packet ingresses a port, a decision is made to sample it or not for cflowd.
2. The original packet is processed for forwarding as normal and the cflowd sample is sent for processing. If a packet is discarded because of filters actions, an indicator is sent with the cflowd sample to the processing agent.
3. If a new flow is found, a new entry is added to the cache. If the flow already exists in the cache, the flow statistics are updated.
4. If a new flow is detected and the maximum number of entries are already in the flow cache, the earliest expiry entry is removed. The earliest expiry entry/flow is the next flow that expires because of the active or inactive timer expiration.
5. If a flow has been inactive for a period of time equal to or greater than the inactive timer (default 15 s), the entry is removed from the flow cache.
6. If a flow has been active for a period of time equal to or greater than the active timer (default 30 min), the entry is removed from the flow cache.

When a flow is exported from the cache, the collected data is sent to an external collector, which maintains an accumulation of historical data flows that network users can use to analyze traffic patterns.

Data is exported in one of the following formats:

- **Version 5**

  Generates a fixed export record for each individual flow captured.

- **Version 8**

  Aggregates multiple individual flows into a fixed aggregate record.

- **Version 9**

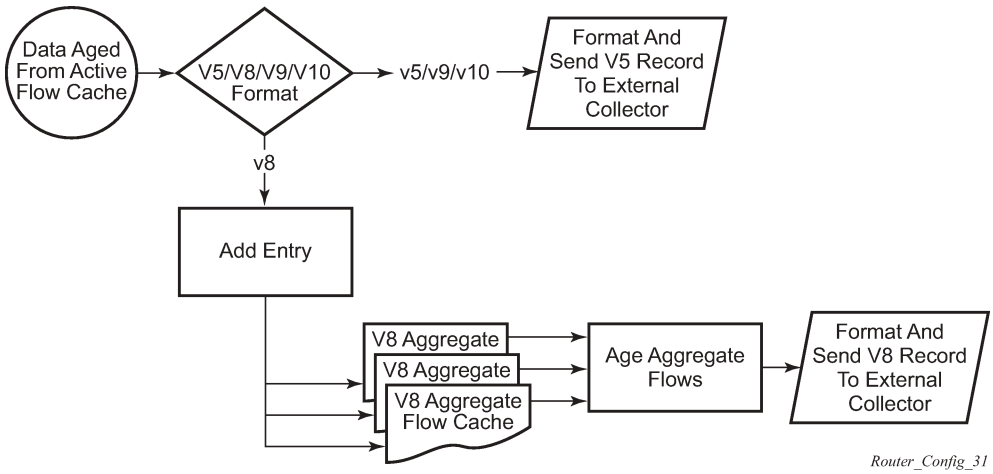
Generates a variable export record, depending on user configuration and sampled traffic type (IPv4, IPv6, or MPLS), for each individual flow captured.

- **Version 10 (IPFIX)**

  Generates a variable export record, depending on user configuration and sampled traffic type (IPv4, IPv6, or MPLS), for each individual flow captured.

Figure 35: V5, V8, V9, V10, and flow processing shows V5, V8, V9, and V10 flow processing.

*Figure 35: V5, V8, V9, V10, and flow processing*




As flows are expired from the active flow cache, the export format must be determined, either V5, V8, V9, and V10.

- If the export format is V5 or V9 and V10, no further processing is performed and the flow data is accumulated to be sent to the external collector.
- If the export format is V8, the flow entry is added to one or more of the configured aggregation matrices.
- As the entries within the aggregate matrices are aged out, they are accumulated to be sent to the external flow collector in V8 format.

The sample rate and cache size are configurable values. The cache size default is 64K flow entries.

A flow terminates when one of the following conditions is met:

- When the inactive timeout period expires (default: 15 s). A flow is considered terminated when no packets are seen for the flow for *n* seconds.
- When an active timeout expires (default: 30 s). Default active timeout is 30 min. A flow terminates according to the time duration, regardless of whether there are packets coming in for the flow.
- When the user executes a **clear cflowd** command.
- When other measures are met that apply to aggressively age flows as the cache becomes too full (such as overflow percent).

### 6.1.1.1 Version 8

There are several different aggregate flow types including:

- AS matrix
- destination prefix matrix
- source prefix matrix
- prefix matrix
- protocol/port matrix

Version 8 is an aggregated export format. As individual flows are aged out of the raw flow cache, the data is added to the aggregate flow cache for each configured aggregate type. Each of these aggregate flows are also aged in a manner similar to the method the active flow cache entries are aged. When an aggregate flow is aged out, it is sent to the external collector in the V8 record format.

#### 6.1.1.2 Version 9

Version 9 format is a more flexible format and allows for different templates or sets of cflowd data to be sent based on the type of traffic being sampled and the template set configured.

Version 9 is interoperable with RFC 3954, *Cisco Systems NetFlow Services Export Version 9*.

#### 6.1.1.3 Version 10

Version 10 is a new format and protocol that interoperates with the specifications from the IETF as the IP Flow Information Export (IPFIX) standard. Like V9, the V10 format uses templates to allow for different data elements about a flow that is to be exported and to handle different type of data flows, such as IPv4, IPv6, and MPLS.

Version 10 is interoperable with RFC 5101 and 5102.

### 6.1.2 Cflowd filter matching

In the filter-matching process, usually every packet is matched against filter (access list) criteria to determine acceptability. With cflowd, only the first packet of a flow is checked. If the first packet is forwarded, an entry is added to the cflowd cache. Subsequent packets in the same flow are then forwarded without needing to be matched against the complete set of filters. Specific performance varies depending on the number and complexity of the filters.

### 6.1.3 Cflowd Collector flow direction configuration

The Cflowd Collector feature allows users to configure the direction of flows sent to the associated Cflowd Collector as **ingress**, **egress**, or **both**. Use the following options to configure the flow direction:

- **both** – flows ingressing or egressing the specified interface match the collector filter (default)
- **ingress** – flows ingressing the specified interface match the collector filter
- **egress** – flows egressing the specified interface match the collector filter

Use the following commands to enable this feature:

- **MD-CLI**

```
configure cflowd collector export-filter interface-list router interface-name direction
configure cflowd collector export-filter interface-list service ies-group-interface
 direction
configure cflowd collector export-filter interface-list service ies-interface direction
configure cflowd collector export-filter interface-list service vprn-group-interface
 direction
configure cflowd collector export-filter interface-list service vprn-interface direction
configure cflowd collector export-filter interface-list service vprn-network-interface
 direction
```

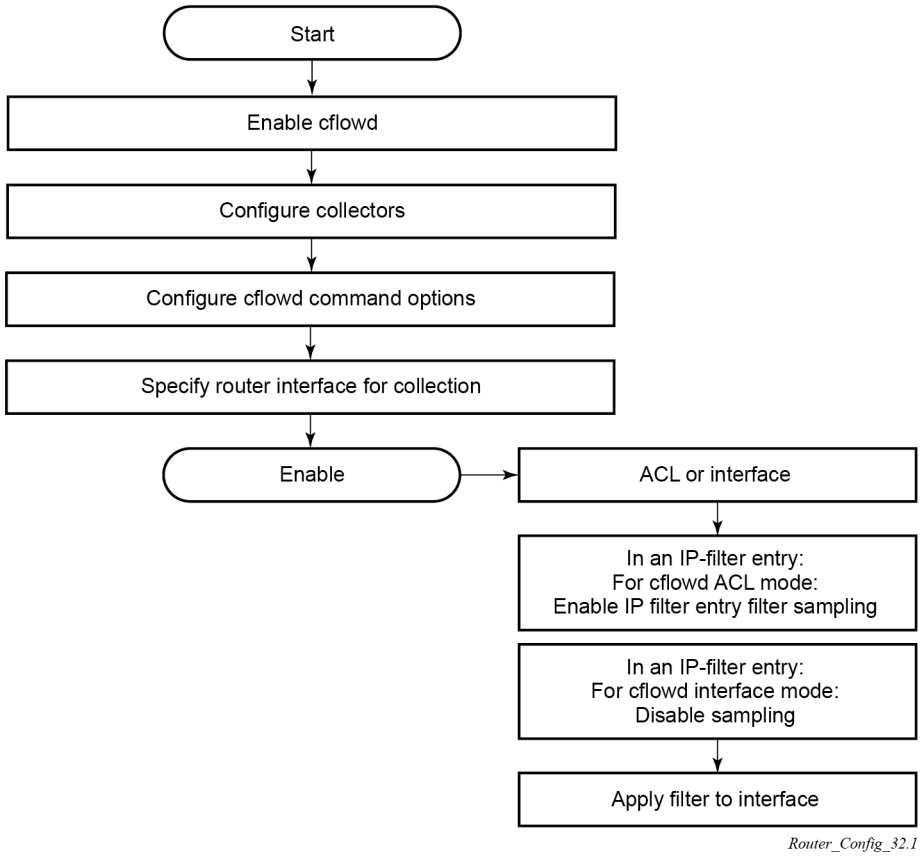
- **classic CLI**

```
configure cflowd collector export-filter interface-list router interface direction
 [direction {ingress | egress | both}]
configure cflowd collector export-filter interface-list service ies subscriber-interface
 group-interface [direction {ingress | egress | both}]
configure cflowd collector export-filter interface-list service ies interface [direction
 {ingress | egress | both}]
configure cflowd collector export-filter interface-list service vprn subscriber-interface
 group-interface [direction {ingress | egress | both}]
configure cflowd collector export-filter interface-list service vprn interface [direction
 {ingress | egress | both}]
configure cflowd collector export-filter interface-list service vprn network-interface
 [direction {ingress | egress | both}]
```

## 6.2 Cflowd configuration process overview

Figure 36: Cflowd configuration and implementation flow shows the process to configure cflowd command options.

*Figure 36: Cflowd configuration and implementation flow*




There are three modes in which cflowd can be enabled to sample traffic on an interface:

- Cflowd interface – where all traffic entering a specified port is subjected to sampling at the configured sampling rate
- Cflowd interface plus – the definition of IP filters that specify an action to disable sampling, where traffic that matches these filter entries is not subject to cflowd sampling

  Use the following commands to disable sampling as part of the IP filter configuration:

  – **MD-CLI**

  ```
  configure filter ip-filter entry interface-sample false
  configure filter ipv6-filter entry interface-sample false
  ```

  – **classic CLI**

  ```
  configure filter ip-filter entry interface-disable-sample
  configure filter ipv6-filter entry interface-disable-sample
  ```

- Cflowd ACL – where IP filters must be created with entries containing the action filter-sampled. In this mode, only traffic matching these filter entries is subject to the cflowd sampling process.

## 6.3 Configuration notes

The following cflowd components must be configured for cflowd to be operational:

- Cflowd is enabled globally.
- At least one collector must be configured and enabled.
- A cflowd option must be specified and enabled on a router interface.
- Sampling must be enabled on either:
  - An IP filter that is applied to a port or service.
  - An interface on a port or service.

## 6.4 Configuring cflowd with CLI

This section provides information to configure cflowd using the command line interface (CLI).

### 6.4.1 Cflowd configuration overview

SR OS implementation of cflowd supports the option to analyze traffic flow. The implementation also supports the use of traffic or access list (ACL) filters to limit the type of traffic that is analyzed.

#### 6.4.1.1 Traffic sampling

Traffic sampling does not examine all packets received by a router. Command options allow the rate at which traffic is sampled and sent for flow analysis to be modified. The default sampling rate is every 1000th packet. Excessive sampling over an extended period of time, for example, more than every 1000th packet, can burden router processing resources.

The following data is maintained for each individual flow in the raw flow cache:

- source IP address
- destinations IP address
- source port
- destination port
- forwarding status
- input interface
- output interface
- IP protocol
- TCP flags
- first timestamp (of the first packet in the flow)
- last timestamp (timestamp of last packet in the flow before the expiry of the flow)
- source AS number for peer and origin (taken from BGP)

- destination AS number for peer and origin (taken from BGP)
- IP next hop
- BGP next hop
- ICMP type and code
- IP version
- source prefix (from routing)
- destination prefix (from routing)
- MPLS label stack from label 1 to 6

Within the raw flow cache, the following characteristics are used to identify an individual flow:

- ingress interface
- source IP address
- destination IP address
- source transport port number
- destination transport port number
- IP protocol type
- IP ToS byte
- virtual router ID
- ICMP type and code
- direction
- MPLS labels

SR OS implementation allows cflowd to be enabled at the interface level or as an action to a filter. By enabling cflowd at the interface level, all IP packets forwarded by the interface are subject to cflowd analysis. By setting cflowd as an action in a filter, only packets matching the specified filter are subject to cflowd analysis. This provides the network user greater flexibility in the types of flows that are captured.

### 6.4.1.2 Collectors

A collector defines how data flows should be exported from the flow cache. A maximum of eight collectors can be configured. Each collector is identified by a unique IP address and UDP port value. Each collector can only export traffic in one version type: V5, V8, V9, or V10.

The command options within a collector configuration can be modified or the defaults retained.

The **autonomous-system-type** command defines whether the autonomous system information to be included in the flow data is based on the originating AS or external peer AS of the flow.

#### 6.4.1.2.1 Aggregation

V8 aggregation allows for flow data to be aggregated into larger, less granular flows. Use aggregation commands to specify the type of data to be collected. These aggregation types are only applicable to flows being exported to a V8 collector.

The following aggregation schemes are supported:

- **AS matrix**

  Flows are aggregated based on source and destination AS and ingress and egress interface.
- **protocol port**

  Flows are aggregated based on the IP protocol, source port number, and destination port number.
- **source prefix**

  Flows are aggregated based on source prefix and mask, source AS, and ingress interface.
- **destination prefix**

  Flows are aggregated based on destination prefix and mask, destination AS, and egress interface.
- **source-destination prefix**

  Flows are aggregated based on source prefix and mask, destination prefix and mask, source and destination AS, ingress interface, and egress interface.
- **raw**

  Flows are not aggregated and are sent to the collector in a V5 record.

### 6.4.2 Basic cflowd configuration

This section provides information to configure cflowd and examples of common configuration tasks. To sample traffic, the following command options must be configured, as a minimum.

- Cflowd must be enabled.
- At least one collector must be configured and enabled.
- Sampling must be enabled on either:
  - an IP filter entry (and applied to a service or a port)
  - an interface applied to a port

The following example shows the cflowd configuration.

**Example: MD-CLI**

```
[ex:/configure cflowd]
A:admin@node-2# info detail
## apply-groups
 ## apply-groups-exclude
    admin-state enable
    analyze-gre-payload false
    analyze-l2tp-traffic false
    analyze-v4overv6-traffic false
    cache-size 6553
    export-mode automatic
    inband-collector-export-only false
    overflow 1
    template-retransmit 600
    use-vrtr-if-index false
    active-flow-timeout 1800
    inactive-flow-timeout 15
    sample-profile 1 {
     ## apply-groups
     ## apply-groups-exclude
        sample-rate 1000
```

```
    }
```

**Example: classic CLI**

```
A:node-2>config>cflowd# info detail
----------------------------------------------
        active-flow-timeout 1800
        cache-size 6553
        inactive-flow-timeout 15
        export-mode automatic
        overflow 1
        template-retransmit 600
        no use-vrtr-if-index
        no inband-collector-export-only
        no analyze-gre-payload
        no analyze-l2tp-traffic
        no analyze-v4overv6-traffic
        sample-profile 1 create
            sample-rate 1000
        exit
        no shutdown
----------------------------------------------
```

### 6.4.3 Common configuration tasks

This section provides a brief overview of the tasks that must be performed to configure cflowd and provides the CLI commands. To begin traffic flow sampling, cflowd must be enabled and at least one collector must be configured.

#### 6.4.3.1 Global cflowd components

The following common (global) attributes apply to all instances of cflowd:

- **active flow timeout**

  The active flow timeout attribute controls the maximum time a flow record can be active before it is automatically exported to defined collectors.

- **inactive flow timeout**

  The inactive flow timeout attribute controls the minimum time before a flow is declared inactive. If no traffic is sampled for a flow for the inactive timeout duration, the flow is declared inactive and marked to be exported to the defined collectors.

- **cache size**

  The cache size attribute defines the maximum size of the flow cache.

- **overflow**

  The overflow attribute defines the percentage of flow records that are exported to all collectors if the flow cache size is exceeded.

- **rate**

  The rate attribute defines the system-wide sampling rate for cflowd.

- **template retransmit**

The template retransmit attribute defines the interval (in seconds) at which the V9 and V10 templates are retransmitted to all configured V9 or V10 collectors.

### 6.4.3.2 Enabling cflowd

Cflowd is disabled by default. Cflowd must be configured with at least one collector to be active. Executing the **cflowd** command enables cflowd.

The following example shows the defaults when cflowd is initially enabled. No collectors or collector options are configured.

**Example: MD-CLI**

```
[ex:/configure cflowd]
A:admin@node-2# info detail
...
    admin-state enable
...
    cache-size 65536
...
     overflow 1
...
    template-retransmit 600
...
    active-flow-timeout 1800
    inactive-flow-timeout 15
    sample-profile 1 {
...
        sample-rate 1000
    }
```

**Example: classic CLI**

```
A:node-2>config>cflowd# info detail
----------------------------------------------
        active-flow-timeout 1800
        cache-size 65536
        inactive-flow-timeout 15
...
        overflow 1
        template-retransmit 600
...
        sample-profile 1 create
            sample-rate 1000
        exit
        no shutdown
----------------------------------------------
```

### 6.4.3.3 Configuring global cflowd

The following example shows the global cflowd configuration.

**Example: MD-CLI**

```
[ex:/configure cflowd]
A:admin@node-2# info
...
```

```
    overflow 10
...
    active-flow-timeout 1800
    inactive-flow-timeout 10
    sample-profile 1 {
        sample-rate 100
    }
```

**Example: classic CLI**

```
A:node-2>config>cflowd# info detail
----------------------------------------------
        active-flow-timeout 1800
...
A:node-2>config>cflowd# info
----------------------------------------------
        inactive-flow-timeout 10
...
        overflow 10
        sample-profile 1 create
            sample-rate 100
        exit
----------------------------------------------
```

### 6.4.3.4 Configuring cflowd collectors

The following example shows a basic configuration of cflowd collectors.

**Example: Basic cflowd collector configuration (MD-CLI)**

```
[ex:/configure cflowd]
A:admin@node-2# info
...
    overflow 10
...
    active-flow-timeout 1800
    inactive-flow-timeout 10
    sample-profile 1 {
        sample-rate 100
    }
    collector 10.10.10.1 port 2000 {
        description "AS info collector"
        version 8
        aggregation {
            as-matrix true
            raw true
        }
    }
 collector 10.10.10.2 port 5000 {
        description "Neighbor collector"
        autonomous-system-type peer
        version 8
        aggregation {
            protocol-port true
            source-destination-prefix true
        }
    }
```

**Example: Basic cflowd collector configuration (classic CLI)**

```
A:node-2>config>cflowd# info detail
----------------------------------------------
        active-flow-timeout 1800
...
A:node-2>config>cflowd# info
----------------------------------------
        inactive-flow-timeout 10
        overflow 10
        sample-profile 1 create
            sample-rate 100
        exit
        collector 10.10.10.1:2000 version 8
            description "AS info collector"
            aggregation
                as-matrix
                raw
            exit
        exit
        collector 10.10.10.2:5000 version 8
            description "Neighbor collector"
            aggregation
                protocol-port
                source-destination-prefix
            exit
            autonomous-system-type peer
        exit
```

**Example: Version 9 collector configuration (MD-CLI)**

```
[ex:/configure cflowd]
A:admin@node-2# info
...
    collector 10.10.10.9 port 2000 {
        description "v9collector"
        template-set mpls-ip
        version 9
    }
```

**Example: Version 9 collector configuration (classic CLI)**

```
A:node-2>config>cflowd# info
----------------------------------------------
...
        collector 10.10.10.9:2000 version 9
            description "v9collector"
            template-set mpls-ip
        exit
----------------------------------------------
```

#### 6.4.3.4.1 Version 9 and Version 10 templates

If the collector is configured to use either V9 or V10 (IPFIX) formats, the flow data is sent to the designated collector using one of the predefined templates. The template used is based on the type of flow for which the data was collected (IPv4, IPv6, MPLS, or Ethernet [Layer 2]), and the configuration of the **template-set** command. The following table lists these options and the corresponding template used to export the flow data.

*Table 14: Template sets*

| Traffic flow | Basic | MPLS-IP |
|---|---|---|
| IPv4 | Basic IPv4 | MPLS-IPv4 |
| IPv6 | Basic IPv6 | MPLS-IPv6 |
| MPLS | Basic MPLS | MPLS-IP |
| Ethernet | L2-IP | L2-IP |

Each flow exported to a collector configured for either V9 or V10 formats is sent using one of the flow template sets listed in Table 14: Template sets.

Table 15: Basic IPv4 template to Table 22: MPLS transport template list the fields in each template listed in Table 14: Template sets.

*Table 15: Basic IPv4 template*

| Field name | Field ID |
|---|---|
| IPv4 Src Addr | 8 |
| IPv4 Dest Addr | 12 |
| IPv4 Nexthop | 15 |
| BGP Nexthop | 18 |
| Ingress Interface | 10 |
| Egress Interface | 14 |
| Packet Count | 2 |
| Byte Count | 1 |
| Start Time | 22 |
| End Time | 21 |
| Flow Start Milliseconds[2] | 152 |
| Flow End Milliseconds[2] | 153 |
| Src Port | 7 |
| Dest Port | 11 |
| Forwarding Status | 89 |

[2] Only sent to collectors configured for V10 format.

| Field name | Field ID |
|---|---|
| TCP control Bits (Flags) | 6 |
| IPv4 Protocol | 4 |
| IPv4 ToS | 5 |
| IP version | 60 |
| ICMP Type and Code | 32 |
| Direction | 61 |
| BGP Source ASN | 16 |
| BGP Dest ASN | 17 |
| Source IPv4 Prefix Length | 9 |
| Dest IPv4 Prefix Length | 13 |
| Minimum IP Total Length | 25 |
| Maximum IP Total Length | 26 |
| Minimum TTL | 52 |
| Maximum TTL | 53 |
| Multicast Replication Factor | 99 |
| bgpNextAdjacentAsNumber | 128 |
| bgpPrevAdjacentAsNumber | 129 |
| IsMulticast[2] | 206 |
| Ingress VRFID[2] | 234 |
| Egress VRFID[2] | 235 |

*Table 16: MPLS-IPv4 template*

| Field Name | Field ID |
|---|---|
| IPv4 Src Addr | 8 |
| IPv4 Dest Addr | 12 |
| IPv4 Nexthop | 15 |
| BGP Nexthop | 18 |
| Ingress Interface | 10 |

| Field Name | Field ID |
|---|---|
| Egress Interface | 14 |
| Packet Count | 2 |
| Byte Count | 1 |
| Start Time | 22 |
| End Time | 21 |
| Flow Start Milliseconds[2] | 152 |
| Flow End Milliseconds[2] | 153 |
| Src Port | 7 |
| Dest Port | 11 |
| Forwarding Status | 89 |
| TCP control Bits (Flags) | 6 |
| IPv4 Protocol | 4 |
| IPv4 ToS | 5 |
| IP version | 60 |
| ICMP Type & Code | 32 |
| Direction | 61 |
| BGP Source ASN | 16 |
| BGP Dest ASN | 17 |
| Source IPv4 Prefix Length | 9 |
| Dest IPv4 Prefix Length | 13 |
| MPLS Top Label Type | 46 |
| MPLS Top Label IPv4 Addr | 47 |
| MPLS Label 1 | 70 |
| MPLS Label 2 | 71 |
| MPLS Label 3 | 72 |
| MPLS Label 4 | 73 |
| MPLS Label 5 | 74 |

| Field Name | Field ID |
|---|---|
| MPLS Label 6 | 75 |
| MPLS Label 7 | 76 |
| MPLS Label 8 | 77 |
| MPLS Label 9 | 78 |
| MPLS Label 10 | 79 |
| Minimum IP Total Length | 25 |
| Maximum IP Total Length | 26 |
| Minimum TTL | 52 |
| Maximum TTL | 53 |
| Multicast Replication Factor | 99 |
| bgpNextAdjacentAsNumber | 128 |
| bgpPrevAdjacentAsNumber | 129 |
| IsMulticast[2] | 206 |
| Ingress VRFID[2] | 234 |
| Egress VRFID[2] | 235 |

*Table 17: Basic IPv6 template*

| Field Name | Field ID |
|---|---|
| IPv6 Src Addr | 27 |
| IPv6 Dest Addr | 28 |
| IPv6 Nexthop | 62 |
| IPv6 BGP Nexthop | 63 |
| IPv4 Nexthop | 15 |
| IPv4 BGP Nexthop | 18 |
| Ingress Interface | 10 |
| Egress Interface | 14 |
| Packet Count | 2 |
| Byte Count | 1 |

| Field Name | Field ID |
|---|---|
| Start Time | 22 |
| End Time | 21 |
| Flow Start Milliseconds[2] | 152 |
| Flow End Milliseconds[2] | 153 |
| Src Port | 7 |
| Dest Port | 11 |
| Forwarding Status | 89 |
| TCP control Bits (Flags) | 6 |
| Protocol | 4 |
| IPv6 Extension Hdr | 64 |
| IPv6 Next Header | 193 |
| IPv6 Flow Label | 31 |
| ToS | 5 |
| IP version | 60 |
| IPv6 ICMP Type & Code | 139 |
| Direction | 61 |
| BGP Source ASN | 16 |
| BGP Dest ASN | 17 |
| IPv6 Src Mask | 29 |
| IPv6 Dest Mask | 30 |
| Minimum IP Total Length | 25 |
| Maximum IP Total Length | 26 |
| Minimum TTL | 52 |
| Maximum TTL | 53 |
| Multicast Replication Factor | 99 |
| bgpNextAdjacentAsNumber | 128 |
| bgpPrevAdjacentAsNumber | 129 |

| Field Name | Field ID |
|---|---|
| IsMulticast[2] | 206 |
| Ingress VRFID[2] | 234 |
| Egress VRFID[2] | 235 |

*Table 18: MPLS-IPv6 template*

| Field name | Field ID |
|---|---|
| IPv6 Src Addr | 27 |
| IPv6 Dest Addr | 28 |
| IPv6 Nexthop | 62 |
| IPv6 BGP Nexthop | 63 |
| IPv4 Nexthop | 15 |
| IPv4 BGP Nexthop | 18 |
| Ingress Interface | 10 |
| Egress Interface | 14 |
| Packet Count | 2 |
| Byte Count | 1 |
| Start Time | 22 |
| End Time | 21 |
| Flow Start Milliseconds[2] | 152 |
| Flow End Milliseconds[2] | 153 |
| Src Port | 7 |
| Dest Port | 11 |
| Forwarding Status | 89 |
| TCP control Bits (Flags) | 6 |
| Protocol | 4 |
| IPv6 Extension Hdr | 64 |
| IPv6 Next Header | 193 |
| IPv6 Flow Label | 31 |

| Field name | Field ID |
|---|---|
| ToS | 5 |
| IP version | 60 |
| IPv6 ICMP Type & Code | 139 |
| Direction | 61 |
| BGP Source ASN | 16 |
| BGP Dest ASN | 17 |
| IPv6 Src Mask | 29 |
| IPv6 Dest Mask | 30 |
| MPLS Top Label Type | 46 |
| MPLS Top Label IPv6 Addr | 47 |
| MPLS Label 1 | 70 |
| MPLS Label 2 | 71 |
| MPLS Label 3 | 72 |
| MPLS Label 4 | 73 |
| MPLS Label 5 | 74 |
| MPLS Label 6 | 75 |
| MPLS Label 7 | 76 |
| MPLS Label 8 | 77 |
| MPLS Label 9 | 78 |
| MPLS Label 10 | 79 |
| MPLS_TOP_LABEL_TYPE | 46 |
| MPLS_TOP_LABEL_ADDR | 47 |
| Minimum IP Total Length | 25 |
| Maximum IP Total Length | 26 |
| Minimum TTL | 52 |
| Maximum TTL | 53 |
| Multicast Replication Factor | 99 |
| bgpNextAdjacentAsNumber | 128 |

| Field name | Field ID |
|---|---|
| bgpPrevAdjacentAsNumber | 129 |
| IsMulticast[2] | 206 |
| Ingress VRFID[2] | 234 |
| Egress VRFID[2] | 235 |

*Table 19: Basic MPLS template*

| Field name | Field ID |
|---|---|
| Start Time | 22 |
| End Time | 21 |
| Flow Start Milliseconds[2] | 152 |
| Flow End Milliseconds[2] | 153 |
| Ingress Interface | 10 |
| Egress Interface | 14 |
| Packet Count | 2 |
| Byte Count | 1 |
| Direction | 61 |
| MPLS Top Label Type | 46 |
| MPLS Top Label Address | 47 |
| MPLS Label 1 | 70 |
| MPLS Label 2 | 71 |
| MPLS Label 3 | 72 |
| MPLS Label 4 | 73 |
| MPLS Label 5 | 74 |
| MPLS Label 6 | 75 |

*Table 20: MPLS-IP template*

| Field name | Field ID |
|---|---|
| IPv4 Src Addr | 8 |

| Field name | Field ID |
|---|---|
| IPv4 Dest Addr | 12 |
| IPv4 Nexthop | 15 |
| IPv6 Src Addr | 27 |
| IPv6 Dest Addr | 28 |
| IPv6 Nexthop | 62 |
| Ingress Interface | 10 |
| Egress Interface | 14 |
| Packet Count | 2 |
| Byte Count | 1 |
| Start Time | 22 |
| End Time | 21 |
| Flow Start Milliseconds[2] | 152 |
| Flow End Milliseconds[2] | 153 |
| Src Port | 7 |
| Dest Port | 11 |
| TCP control Bits (Flags) | 6 |
| IPv4 Protocol | 4 |
| IPv4 ToS | 5 |
| IP version | 60 |
| ICMP Type & Code | 32 |
| IPv6 Flow Label | 31 |
| Direction | 61 |
| MPLS Top Label Type | 46 |
| MPLS Top Label IPv4 Addr | 47 |
| MPLS Label 1 | 70 |
| MPLS Label 2 | 71 |
| MPLS Label 3 | 72 |

| Field name | Field ID |
|---|---|
| MPLS Label 4 | 73 |
| MPLS Label 5 | 74 |
| MPLS Label 6 | 75 |
| MPLS Label 7 | 76 |
| MPLS Label 8 | 77 |
| MPLS Label 9 | 78 |
| MPLS Label 10 | 79 |

To address Table 21: L2-IP (Ethernet) flow template, only one Ethernet (L2-IP) flow template is supported and exported to IPFIX (V10) collectors.

*Table 21: L2-IP (Ethernet) flow template*

| Field name | Field ID |
|---|---|
| MAC Src Addr | 56 |
| MAC Dest Addr | 80 |
| Ingress Physical Interface | 252 |
| Egress Physical Interface | 253 |
| Dot1q VLAN ID | 243 |
| Dot1q Customer VLAN ID | 245 |
| Post Dot1q VLAN ID | 254 |
| Post Dot1q Customer VLAN Id | 255 |
| IPv4 Src Addr | 8 |
| IPv4 Dest Addr | 12 |
| IPv6 Src Addr | 27 |
| IPv6 Dest Addr | 28 |
| Packet Count | 2 |
| Byte Count | 1 |
| Flow Start Milliseconds | 152 |
| Flow End Milliseconds | 153 |

| Field name | Field ID |
|---|---|
| Src Port | 7 |
| Dest Port | 11 |
| TCP control Bits (Flags) | 6 |
| Protocol | 4 |
| IPv6 Option Header | 64 |
| IPv6 Next Header | 196 |
| IPv6 Flow Label | 31 |
| ToS | 5 |
| IP Version | 60 |
| ICMP Type Code | 32 |
| Ingress VRF | 234 |
| IPv4 BGP Nexthop | 18 |
| IPv6 BGP Nexthop | 63 |

*Table 22: MPLS transport template*

| Field name | Field ID |
|---|---|
| Flow Start Milliseconds | 152 |
| Flow End Milliseconds | 153 |
| VRF ID | 234 |
| Ingress Interface | 10 |
| Packet Count | 2 |
| Byte Count | 1 |
| Direction | 61 |
| MPLS_TOP_LABEL_TYPE | 46 |
| MPLS_TOP_LABEL_ADDR | 47 |
| MPLS Label-1 | 70 |

*Table 23: GRE flow template*

| Field name[3] | Field ID |
|---|---|
| IPv4 Src Addr[4] | 8 |
| IPv4 Dest Addr[4] | 12 |
| Ingress ID | 252 |
| Egress ID | 253 |
| Flow Start Milliseconds | 152 |
| Ingress VRF ID[4] | 234 |
| Egress VRF ID[4] | 235 |
| Protocol[4] | 4 |
| ToS[4] | 5 |
| Data Link Frame Size[5] | 312 |
| Section Exported Octets[4] | 410 |
| Data Link Frame Section[4] | 315 |

### 6.4.3.5 Specifying cflowd on an IP interface

When cflowd is enabled on an interface, all packets forwarded by the interface are subject to analysis according to the global cflowd configuration and sorted according to the collector configurations.

See Table 30: Cflowd configuration dependencies for configuration combinations.

Use the following command to configure cflowd on an IP interface.

```
configure router interface cflowd-parameters sampling unicast type interface
```

When the preceding command is configured, the following requirements must be met to enable traffic sampling on the interface:

- Enable cflowd.
- Ensure at least one cflowd collector is configured and enabled.

3 The field names are exported only to IPFIX (V10) collectors.

4 The IP fields contain values from the outer GRE IP header.

5 The Data Link Frame section field includes the inner IP headers.

- Use the commands in the following context to configure sampling as unicast or multicast, as well as the type and direction of the sampling. By default, the direction is **ingress-only**.

```
configure router interface cflowd-parameters sampling
```

- Use the following commands to prevent specific types of traffic from being sampled when interface sampling is enabled. The filter must be applied to the service or network interface on which the traffic to be omitted is to ingress the system.
  - **MD-CLI**

```
configure filter ip-filter entry interface-sample false
configure filter ipv6-filter entry interface-sample false
```

  - **classic CLI**

```
configure filter ip-filter entry interface-disable-sample
configure filter ipv6-filter entry interface-disable-sample
```

#### 6.4.3.5.1 Interface sampling configuration

Use the commands in the following context to configure cflowd sampling on an interface.

```
configure router interface cflowd-parameters sampling
```

Depending on the sampling type command option selected, either **acl** or **interface**, cflowd extracts traffic flow samples from an IP filter or an interface for analysis. All packets forwarded by the interface are analyzed according to the cflowd configuration.

The **acl** command option must be selected to enable traffic sampling on an IP filter. Cflowd must be enabled in at least one IP filter entry. Use the following command to enable cflowd sampling on an IP filter entry:

- **MD-CLI**

```
configure filter ip-filter entry filter-sample true
```

- **classic CLI**

```
configure filter ip-filter entry filter-sample
```

The **interface** command option must be selected to enable traffic sampling on an interface. If cflowd is not enabled, traffic sampling does not occur on the interface.

#### 6.4.3.5.2 Service interfaces

Use the commands in the following context to configure cflowd on a service interface.

```
configure router interface cflowd-parameters sampling
```

When enabled on a service interface, cflowd collects routed traffic flow samples through a router for analysis. Cflowd is supported on IES and VPRN service interfaces only. Layer 2 traffic is excluded. All packets forwarded by the interface are analyzed according to the cflowd configuration. On the interface

level, cflowd can be associated with a filter (ACL) or an IP interface. Layer 2 cflowd ingress sampling is supported on VPLS and Epipe SAPs.

#### 6.4.3.5.2.1 Compact templates

*Table 24: IPv4 flow record*

| IPFIX Field | Field ID |
|---|---|
| Packet | 2 |
| Byte | 1 |
| Input ifIndex | 10 |
| Output ifIndex | 14 |
| IP version | 60 |
| IP Src Port | 7 |
| IP Dst Port | 11 |
| IP proto | 4 |
| IP tcpflags | 6 |
| Flow Start | 22/152 |
| Flow Stop | 21/153 |
| IP min TTL | 52 |
| IP max TTL | 53 |
| IP tos | 5 |
| Flow Direction | 61 |
| IP icmp type/code | 32 |
| Forwarding status | 89 |
| IP src Address | 8 (IPv4) |
| IP dst Address | 12 (IPv4) |

*Table 25: IPv6 flow record*

| IPFIX Field | Field ID |
|---|---|
| Packet | 2 |
| Byte | 1 |

| IPFIX Field | Field ID |
|---|---|
| Input ifIndex | 10 |
| Output ifIndex | 14 |
| IP version | 60 |
| IP Src Port | 7 |
| IP Dst Port | 11 |
| IP proto | 4 |
| IP tcpflags | 6 |
| Flow Start | 22/152 |
| Flow Stop | 21/153 |
| IP min TTL | 52 |
| IP max TTL | 53 |
| IP tos | 5 |
| Flow Direction | 61 |
| IPv6 ICMP type/code | 139 |
| Forwarding status | 89 |
| IP src Address | 27(IPv6) |
| IP dst Address | 28(IPv6) |

*Table 26: MPLS flow record (v9 and v10)*

| IPFIX Field | Field ID |
|---|---|
| Flow Start | 22/152 |
| Flow Stop | 21/153 |
| Input ifIndex | 10 |
| Output ifIndex | 14 |
| Packet | 2 |
| Byte | 1 |
| Flow Direction | 61 |
| MPLS Top Label | 46 |

| IPFIX Field | Field ID |
|---|---|
| MPLS Top Label IPv4 Address | 47 |
| MPLS Label 1 | 70 |
| MPLS Label 2 | 71 |
| MPLS Label 3 | 72 |
| MPLS Label 4 | 73 |
| MPLS Label 5 | 74 |
| MPLS Label 6 | 75 |
| MPLS Label 7 | 76 |
| MPLS Label 8 | 77 |
| MPLS Label 9 | 78 |
| MPLS Label 10 | 79 |

*Table 27: Layer-2 flow record (v10 only)*

| IPFIX Field | Field ID |
|---|---|
| Source MAC Address | 53 |
| Destination MAC Address | 80 |
| Ingress Physical Interface | 252 |
| Egress Physical Interface | 253 |
| Dot1q VLAN ID | 243 |
| Dot1q Customer VLAN ID | 245 |
| Post Dot1q VLAN ID | 254 |
| Post Dot1q Customer VLAN ID | 255 |
| IPv4 src Address | 8 |
| IPv4 dst Address | 12 |
| IPv6 src Address | 27 |
| IPv6 dst Address | 28 |
| Packet Count | 2 |
| Byte Count | 1 |

| IPFIX Field | Field ID |
|---|---|
| Flow Start Millisecond | 152 |
| Flow End Millisecond | 153 |

#### 6.4.3.5.2.2 FP-accelerated templates

*Table 28: IPv4 flow record*

| IPFIX Field | Field ID |
|---|---|
| Byte | 1 |
| Input ifIndex | 10 |
| Output ifIndex | 14 |
| IP version | 60 |
| IP src Port | 7 |
| IP Dst Port | 11 |
| IP Proto | 4 |
| IP TCP flags | 6 |
| IP min TTL | 52 |
| IP TOS | 5 |
| IP icmp type/code | 32 |
| Forwarding status | 89 |
| IP src Address1 | 8 (IPv4) |
| IP dst Address1 | 12 (IPv4) |

*Table 29: IPv6 flow record*

| IPFIX Field | Field ID |
|---|---|
| Byte | 1 |
| Input ifIndex | 10 |
| Output ifIndex | 14 |
| IP version | 60 |
| IP src Port | 7 |

| IPFIX Field | Field ID |
|---|---|
| IP Dst Port | 11 |
| IP Proto | 4 |
| IP TCP flags | 6 |
| IP min TTL | 52 |
| IP TOS | 5 |
| Forwarding status | 89 |
| IP src Address1 | 27(IPv6) |
| IP dst Address1 | 28(IPv6) |

### 6.4.3.6 Specifying sampling options in filter entries

Packets are matched against filter entries to determine acceptability. With cflowd, only the first packet of a flow is compared. If the first packet matches the filter criteria, an entry is added to the cflowd cache. Subsequent packets in the same flow are also sampled based on the cache entry.

Because a filter can be applied to more than one interface (when configured with a scope template), you can enable or disable traffic sampling on an interface-by-interface basis. Use the following command to disable traffic sampling:

- **MD-CLI**

```
configure filter ip-filter entry interface-sample false
```

- **classic CLI**

```
configure filter ip-filter entry interface-disable-sample
```

The preceding command to disable traffic sampling can be enabled or disabled as needed instead of having to create numerous filter versions.

To enable an interface for filter traffic sampling, the following requirements must be met:

- Cflowd must be enabled globally.
- At least one cflowd collector must be configured and enabled.
- Use the commands in the following context on the IP interface that is used to configure sampling as unicast or multicast. You must also select the ACL option.

```
configure router interface cflowd-parameters sampling
```

- On the IP filter being used, you must explicitly enable filter sampling for the entries matching the traffic that should be sampled. Use the following commands to configure filter sampling for the filter:
  - **MD-CLI**

```
configure filter ip-filter entry filter-sample true
```

```
configure filter ipv6-filter entry filter-sample true
```

– **classic CLI**

```
configure filter ip-filter entry filter-sample
configure filter ipv6-filter entry filter-sample
```

The default is disabled. See Filter configurations for more information.

- The filter must be applied to a service or a network interface. The service or port must be enabled and operational.

#### 6.4.3.6.1 Filter configurations

When a filter policy is applied to a service or a network interface, sampling can be configured so that traffic matching the associated IP filter entry is sampled when the IP interface is set to cflowd ACL mode and filter sampling is enabled. Use the following command to enable cflowd sampling on an IP filter entry:

- **MD-CLI**

```
configure filter ip-filter entry filter-sample true
```

- **classic CLI**

```
configure filter ip-filter entry filter-sample
```

When the traffic sampling is disabled, traffic matching the associated IP filter entry is not sampled if the IP interface is set to cflowd ACL mode. Use the following command to disable traffic sampling:

- **MD-CLI**

```
configure filter ip-filter entry interface-sample false
```

- **classic CLI**

```
configure filter ip-filter entry interface-disable-sample
```

#### 6.4.3.6.2 Dependencies

For cflowd to be operational, the following requirements must be met:

- Cflowd must be enabled on a global level. If cflowd is disabled, any traffic sampling instances are also disabled.
- At least one collector must be configured and enabled in order for traffic sampling to occur on an enabled entity.
- If a specific collector UDP port is not identified, flows are sent to port 2055 by default.

Cflowd can also be dependent on the following entity configurations:

- Interface sampling configuration
- Service interfaces
- Filter configurations

The combination of interface and filter entry configurations determines whether flow sampling occurs. Table 30: Cflowd configuration dependencies lists the expected results based on cflowd configuration dependencies.

*Table 30: Cflowd configuration dependencies*

| Interface Setting | cflowd-parameter type Setting | Command ip-filter *entry* Setting | Expected Results |
|---|---|---|---|
| IP-filter mode | ACL | **filter-sample true** (MD-CLI)<br>**filter-sample** (classic CLI) | Traffic matching is sampled at specified rate |
| IP-filter mode | ACL | **filter-sample false** (MD-CLI)<br>**no filter-sample** (classic CLI) | No traffic is sampled on this interface |
| IP-filter mode or cflowd not enabled on interface | ACL | **interface-sample false** (MD-CLI)<br>**interface-disable-sample** (classic CLI) | Command is ignored; no sampling occurs |
| Interface mode | Interface | **interface-sample false** (MD-CLI)<br>**interface-disable-sample** (classic CLI) | Traffic matching this IP filter entry is not sampled |
| Interface mode | Interface | none | All IP traffic ingressing the interface is subject to sampling |
| Interface mode | Interface | **filter-sample true** (MD-CLI)<br>**filter-sample** (classic CLI) | Filter-level action is ignored; all traffic ingressing the interface is subject to sampling |

### 6.4.3.7 Configuring Cflowd Collector flow direction

The following example shows how to configure the direction of flows to the Cflowd Collector.

**Example: MD-CLI**

```
[ex:/configure cflowd]
A:admin@node-2# info
    overflow 10
    template-retransmit 60
    active-flow-timeout 30
    inactive-flow-timeout 10
    collector 192.168.202.171 port 2055 {
        description "test"
        version 9
        export-filter {
            interface-list {
                service {
                    ies-interface service-name "28000" interface-name "ies-28000" {
                        direction ingress
```

```
                    }
                    ies-interface service-name "33600" interface-name "ies-33600" {
                        direction ingress
                    }
                }
            }
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>cflowd# info
----------------------------------------------
        active-flow-timeout 30
        inactive-flow-timeout 10
        overflow 10
        template-retransmit 60
        collector 192.168.202.171 version 9
            description "test"
            export-filter
                interface-list
                    service
                        ies 28000 interface ies-28000 direction ingress
                        ies 33600 interface ies-33600 direction ingress
                    exit
                exit
            exit
        exit
----------------------------------------------
```

## 6.5 Cflowd configuration management tasks

This section describes cflowd configuration management tasks.

### 6.5.1 Modifying global cflowd

Cflowd modifications apply to all instances where cflowd is enabled. Use the commands in the following context to configure cflowd.

```
configure cflowd
```

**Example: Modification of a cflowd configuration (MD-CLI)**

```
*[ex:/configure cflowd]
A:admin@node-2# active-flow-timeout 3600

*[ex:/configure cflowd]
A:admin@node-2# inactive-flow-timeout 15

*[ex:/configure cflowd]
A:admin@node-2# overflow 2

*[ex:/configure cflowd]
A:admin@node-2# sample-profile 1

*[ex:/configure cflowd sample-profile 1]
```

```
A:admin@node-2# sample-rate 10
```

**Example: Cflowd configuration output (MD-CLI)**

```
[ex:/configure cflowd]
A:admin@node-2# info detail
...
    inactive-flow-timeout 15
...
*[ex:/configure cflowd]
A:admin@node-2# info
...
    overflow 2
...
    active-flow-timeout 3600
    sample-profile 1 {
        sample-rate 10
    }
...
   }
```

**Example: Modification of a cflowd configuration (classic CLI)**

```
*A:node-2>config>cflowd# active-flow-timeout 3600
*A:node-2>config>cflowd# inactive-flow-timeout 15
*A:node-2>config>cflowd# overflow 2
*A:node-2>config>cflowd# sample-profile 1 create
*A:node-2>config>cflowd>sample-profile# sample-rate 10
```

**Example: Cflowd configuration output (classic CLI)**

```
A:node-2>config>cflowd# info detail
----------------------------------------------
 ...
        inactive-flow-timeout 15
...
*A:node-2>config>cflowd# info
----------------------------------------------
        active-flow-timeout 3600
...
        overflow 2
        sample-profile 1 create
            sample-rate 10
        exit
```

### 6.5.2 Modifying cflowd collector command options

Use the commands in the following context to modify cflowd collector and aggregation command options.

```
configure cflowd
```

If a specific collector UDP port is not identified, flows are sent to port 2055 by default.

The following example shows a basic cflowd configuration that has been modified.

**Example: MD-CLI**

```
[ex:/configure cflowd]
```

```
A:admin@node-2# info
...
    overflow 2
...
    active-flow-timeout 3600
    sample-profile 1 {
        sample-rate 10
    }
    collector 10.10.10.1 port 2000 {
        description "AS info collector"
        version 8
        }
    }
    collector 10.10.10.2 port 5000 {
        description "Test collector"
        version 9
        aggregation {
            source-prefix true
            raw true
        }
    }
```

**Example: classic CLI**

```
A:node-2>config>cflowd# info
----------------------------------------
     active-flow-timeout 3600
        overflow 2
        sample-profile 1 create
            sample-rate 10
        exit
        collector 10.10.10.1:2000 version 8
            description "AS info collector"
        exit
        collector 10.10.10.2:5000 version 9
            description "Test collector"
            aggregation
                source-prefix
                raw
            exit
        exit
----------------------------------------
```

## 6.6 FP acceleration for cflowd processing

FP acceleration for cflowd allows the FP complex on specific CPMs to process and directly export IPv4 and IPv6 flow records. This feature supports significantly higher sampling capacity and flow record generation. The feature requires using CPM-2 or later in 7750 SR-7s and SR-14s, and 7950 XRS. When enabled, cflowd samples from configured interfaces are sent to the FP complex located on the CPM, which then pulls specific information from the IPv4 or IPv6 headers to populate the FP, accelerated flow record template. This mechanism generates a flow record for each sample.

### 6.6.1 Configuring FP acceleration for cflowd processing

**Note:** The following information applies for the MD-CLI.

To enable FP-accelerated cflowd processing, configure the following:

- Use the following command to configure a cflowd collector for FP-accelerated cflowd processing.

```
configure cflowd collector template-set fastpath
```

- Use the following command to configure one or more sample profiles.

```
configure cflowd sample-profile metering-process fp-accelerated
```

The following example shows the configuration of FP acceleration for cflowd processing.

**Example: MD-CLI**

```
[ex:/configure]
A:admin@node-2# info
    cflowd {
        admin-state enable
...
        inband-collector-export-only true
...
        sample-profile 2 {
            sample-rate 2000
            metering-process fp-accelerated
        }
        collector 10.10.10.10 port 1 {
            template-set fastpath
            version 10
        }
    }
```

### 6.6.2 Supported forwarding status codes

The following table shows supported forwarding status codes.

*Table 31: Supported forwarding status codes*

| Status | Reported code (field 89) |
|---|---|
| Forwarded | 64 |
| Drop-ACL | 130 |
| Drop-Unroutable | 131 |
| Drop-Fragmentation needed but DF bit set | 133 |
| Drop-uRPF failure | 140 |

# 7 Standards and protocol support

**Note:**
The information provided in this chapter is subject to change without notice and may not apply to all platforms.

Nokia assumes no responsibility for inaccuracies.

## 7.1 Access Node Control Protocol (ANCP)

draft-ietf-ancp-protocol-02, *Protocol for Access Node Control Mechanism in Broadband Networks*

RFC 5851, *Framework and Requirements for an Access Node Control Mechanism in Broadband Multi-Service Networks*

## 7.2 Bidirectional Forwarding Detection (BFD)

draft-ietf-lsr-ospf-bfd-strict-mode-10, *OSPF BFD Strict-Mode*

RFC 5880, *Bidirectional Forwarding Detection (BFD)*

RFC 5881, *Bidirectional Forwarding Detection (BFD) IPv4 and IPv6 (Single Hop)*

RFC 5882, *Generic Application of Bidirectional Forwarding Detection (BFD)*

RFC 5883, *Bidirectional Forwarding Detection (BFD) for Multihop Paths*

RFC 7130, *Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces*

RFC 7880, *Seamless Bidirectional Forwarding Detection (S-BFD)*

RFC 7881, *Seamless Bidirectional Forwarding Detection (S-BFD) for IPv4, IPv6, and MPLS*

RFC 7883, *Advertising Seamless Bidirectional Forwarding Detection (S-BFD) Discriminators in IS-IS*

RFC 7884, *OSPF Extensions to Advertise Seamless Bidirectional Forwarding Detection (S-BFD) Target Discriminators*

RFC 9247, *BGP - Link State (BGP-LS) Extensions for Seamless Bidirectional Forwarding Detection (S-BFD)*

## 7.3 Border Gateway Protocol (BGP)

draft-gredler-idr-bgplu-epe-14, *Egress Peer Engineering using BGP-LU*

draft-hares-idr-update-attrib-low-bits-fix-01, *Update Attribute Flag Low Bits Clarification*

draft-ietf-idr-add-paths-guidelines-08, *Best Practices for Advertisement of Multiple Paths in IBGP*

draft-ietf-idr-best-external-03, *Advertisement of the best external route in BGP*

draft-ietf-idr-bgp-flowspec-oid-03, *Revised Validation Procedure for BGP Flow Specifications*

draft-ietf-idr-bgp-gr-notification-01, *Notification Message support for BGP Graceful Restart*

draft-ietf-idr-bgp-ls-app-specific-attr-16, *Application-Specific Attributes Advertisement with BGP Link-State*

draft-ietf-idr-bgp-ls-flex-algo-06, *Flexible Algorithm Definition Advertisement with BGP Link-State*

draft-ietf-idr-bgp-optimal-route-reflection-10, *BGP Optimal Route Reflection (BGP-ORR)*

draft-ietf-idr-error-handling-03, *Revised Error Handling for BGP UPDATE Messages*

draft-ietf-idr-flowspec-interfaceset-03, *Applying BGP flowspec rules on a specific interface set*

draft-ietf-idr-flowspec-path-redirect-05, *Flowspec Indirection-id Redirect* – localised ID

draft-ietf-idr-flowspec-redirect-ip-02, *BGP Flow-Spec Redirect to IP Action*

draft-ietf-idr-link-bandwidth-03, *BGP Link Bandwidth Extended Community*

draft-ietf-idr-long-lived-gr-00, *Support for Long-lived BGP Graceful Restart*

RFC 1772, *Application of the Border Gateway Protocol in the Internet*

RFC 1997, *BGP Communities Attribute*

RFC 2385, *Protection of BGP Sessions via the TCP MD5 Signature Option*

RFC 2439, *BGP Route Flap Damping*

RFC 2545, *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*

RFC 2858, *Multiprotocol Extensions for BGP-4*

RFC 2918, *Route Refresh Capability for BGP-4*

RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*

RFC 4360, *BGP Extended Communities Attribute*

RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 4456, *BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)*

RFC 4486, *Subcodes for BGP Cease Notification Message*

RFC 4659, *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*

RFC 4684, *Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*

RFC 4724, *Graceful Restart Mechanism for BGP* – helper mode

RFC 4760, *Multiprotocol Extensions for BGP-4*

RFC 4798, *Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*

RFC 5004, *Avoid BGP Best Path Transitions from One External to Another*

RFC 5065, *Autonomous System Confederations for BGP*

RFC 5291, *Outbound Route Filtering Capability for BGP-4*

RFC 5396, *Textual Representation of Autonomous System (AS) Numbers* – asplain

RFC 5492, *Capabilities Advertisement with BGP-4*

RFC 5668, *4-Octet AS Specific BGP Extended Community*

RFC 6286, *Autonomous-System-Wide Unique BGP Identifier for BGP-4*

RFC 6368, *Internal BGP as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 6793, *BGP Support for Four-Octet Autonomous System (AS) Number Space*

RFC 6810, *The Resource Public Key Infrastructure (RPKI) to Router Protocol*

RFC 6811, *Prefix Origin Validation*

RFC 6996, *Autonomous System (AS) Reservation for Private Use*

RFC 7311, *The Accumulated IGP Metric Attribute for BGP*

RFC 7606, *Revised Error Handling for BGP UPDATE Messages*

RFC 7607, *Codification of AS 0 Processing*

RFC 7674, *Clarification of the Flowspec Redirect Extended Community*

RFC 7752, *North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP*

RFC 7854, *BGP Monitoring Protocol (BMP)*

RFC 7911, *Advertisement of Multiple Paths in BGP*

RFC 7999, *BLACKHOLE Community*

RFC 8092, *BGP Large Communities Attribute*

RFC 8097, *BGP Prefix Origin Validation State Extended Community*

RFC 8212, *Default External BGP (EBGP) Route Propagation Behavior without Policies*

RFC 8277, *Using BGP to Bind MPLS Labels to Address Prefixes*

RFC 8571, *BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions*

RFC 8950, *Advertising IPv4 Network Layer Reachability Information (NLRI) with an IPv6 Next Hop*

RFC 8955, *Dissemination of Flow Specification Rules*

RFC 8956, *Dissemination of Flow Specification Rules for IPv6*

RFC 9086, *Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing BGP Egress Peer Engineering*

## 7.4 Bridging and management

IEEE 802.1AB, *Station and Media Access Control Connectivity Discovery*

IEEE 802.1ad, *Provider Bridges*

IEEE 802.1ag, *Connectivity Fault Management*

IEEE 802.1ah, *Provider Backbone Bridges*

IEEE 802.1ak, *Multiple Registration Protocol*

IEEE 802.1aq, *Shortest Path Bridging*

IEEE 802.1AX, *Link Aggregation*

IEEE 802.1D, *MAC Bridges*

IEEE 802.1p, *Traffic Class Expediting*

IEEE 802.1Q, *Virtual LANs*

IEEE 802.1s, *Multiple Spanning Trees*

IEEE 802.1w, *Rapid Reconfiguration of Spanning Tree*

IEEE 802.1X, *Port Based Network Access Control*

## 7.5 Broadband Network Gateway (BNG) Control and User Plane Separation (CUPS)

3GPP TS 23.003, *Numbering, addressing and identification*

3GPP TS 23.007, *Restoration procedures*

3GPP TS 23.402, *Architecture enhancements for non-3GPP accesses* – S2a roaming based on GPRS

3GPP TS 23.501, *System architecture for the 5G System (5GS)*

3GPP TS 23.502, *Procedures for the 5G System (5GS)*

3GPP TS 23.503, *Policy and charging control framework for the 5G System (5GS)*

3GPP TS 24.501, *Non-Access-Stratum (NAS) protocol for 5G System (5GS)*

3GPP TS 29.244, *Interface between the Control Plane and the User Plane nodes*

3GPP TS 29.281, *General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U)*

3GPP TS 29.500, *Technical Realization of Service Based Architecture*

3GPP TS 29.501, *Principles and Guidelines for Services Definition*

3GPP TS 29.502, *Session Management Services*

3GPP TS 29.503, *Unified Data Management Services*

3GPP TS 29.512, *Session Management Policy Control Service*

3GPP TS 29.518, *Access and Mobility Management Services*

3GPP TS 32.255, *5G data connectivity domain charging*

3GPP TS 32.290, *Services, operations and procedures of charging using Service Based Interface (SBI)*

3GPP TS 32.291, *5G system, charging service*

BBF TR-459, *Control and User Plane Separation for a Disaggregated BNG*

BBF TR-459.2, *Multi-Service Disaggregated BNG with CUPS: Integrated Carrier Grade NAT function*

RFC 8300, *Network Service Header (NSH)*

RFC 8910, *Captive-Portal Identification in DHCP and Router Advertisements (RAs)*

## 7.6 Certificate management

RFC 4210, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*

RFC 4211, *Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)*

RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*

RFC 6712, *Internet X.509 Public Key Infrastructure -- HTTP Transfer for the Certificate Management Protocol (CMP)*

RFC 7030, *Enrollment over Secure Transport*

RFC 7468, *Textual Encodings of PKIX, PKCS, and CMS Structures*

## 7.7 Circuit emulation

RFC 4553, *Structure-Agnostic Time Division Multiplexing (TDM) over Packet (SAToP)*

RFC 5086, *Structure-Aware Time Division Multiplexed (TDM) Circuit Emulation Service over Packet Switched Network (CESoPSN)*

RFC 5287, *Control Protocol Extensions for the Setup of Time-Division Multiplexing (TDM) Pseudowires in MPLS Networks*

## 7.8 Ethernet

IEEE 802.3ah, *Media Access Control Parameters, Physical Layers, and Management Parameters for Subscriber Access Networks*

IEEE 802.3x, *Ethernet Flow Control*

ITU-T G.8031/Y.1342, *Ethernet Linear Protection Switching*

ITU-T G.8032/Y.1344, *Ethernet Ring Protection Switching*

ITU-T Y.1731, *OAM functions and mechanisms for Ethernet based networks*

## 7.9 Ethernet VPN (EVPN)

draft-ietf-bess-bgp-srv6-args-00, *SRv6 Argument Signaling for BGP Services*

draft-ietf-bess-evpn-ip-aliasing-00, *EVPN Support for L3 Fast Convergence and Aliasing/Backup Path* – IP Prefix routes

draft-ietf-bess-evpn-ipvpn-interworking-06, *EVPN Interworking with IPVPN*

draft-ietf-bess-evpn-irb-mcast-09, *EVPN Optimized Inter-Subnet Multicast (OISM) Forwarding* – ingress replication and mLDP

draft-ietf-bess-evpn-pref-df-06, *Preference-based EVPN DF Election*

draft-ietf-bess-evpn-unequal-lb-16, *Weighted Multi-Path Procedures for EVPN Multi-Homing* – section 9

draft-ietf-bess-evpn-virtual-eth-segment-06, *EVPN Virtual Ethernet Segment*

draft-ietf-bess-pbb-evpn-isid-cmacflush-00, *PBB-EVPN ISID-based CMAC-Flush*

draft-sr-bess-evpn-vpws-gateway-03, *Ethernet VPN Virtual Private Wire Services Gateway Solution*

RFC 7432, *BGP MPLS-Based Ethernet VPN*

RFC 7623, *Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)*

RFC 8214, *Virtual Private Wire Service Support in Ethernet VPN*

RFC 8317, *Ethernet-Tree (E-Tree) Support in Ethernet VPN (EVPN) an Provider Backbone Bridging EVPN (PBB-EVPN)*

RFC 8365, *A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)*

RFC 8560, *Seamless Integration of Ethernet VPN (EVPN) with Virtual Private LAN Service (VPLS) and Their Provider Backbone Bridge (PBB) Equivalents*

RFC 8584, *DF Election and AC-influenced DF Election*

RFC 9047, *Propagation of ARP/ND Flags in an Ethernet Virtual Private Network (EVPN)*

RFC 9135, *Integrated Routing and Bridging in Ethernet VPN (EVPN)* – Asymmetric IRB Procedures and Mobility Procedure

RFC 9136, *IP Prefix Advertisement in Ethernet VPN (EVPN)*

RFC 9161, *Operational Aspects of Proxy ARP/ND in Ethernet Virtual Private Networks*

RFC 9251, *Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Proxies for Ethernet VPN (EVPN)*

## 7.10 gRPC Remote Procedure Calls (gRPC)

cert.proto version 0.1.0, *gRPC Network Operations Interface (gNOI) Certificate Management Service*

file.proto version 0.1.0, *gRPC Network Operations Interface (gNOI) File Service*

gnmi.proto version 0.8.0, *gRPC Network Management Interface (gNMI) Service Specification*

PROTOCOL-HTTP2, *gRPC over HTTP2*

system.proto Version 1.0.0, *gRPC Network Operations Interface (gNOI) System Service*

## 7.11 Intermediate System to Intermediate System (IS-IS)

draft-ietf-isis-mi-02, *IS-IS Multi-Instance*

draft-ietf-lsr-igp-ureach-prefix-announce-01, *IGP Unreachable Prefix Announcement* – without U-Flag and UP-Flag

draft-kaplan-isis-ext-eth-02, *Extended Ethernet Frame Size Support*

ISO/IEC 10589:2002 Second Edition, *Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)*

RFC 1195, *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*

RFC 2973, *IS-IS Mesh Groups*

RFC 3359, *Reserved Type, Length and Value (TLV) Codepoints in Intermediate System to Intermediate System*

RFC 3719, *Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)*

RFC 3787, *Recommendations for Interoperable IP Networks using Intermediate System to Intermediate System (IS-IS)*

RFC 5120, *M-ISIS: Multi Topology (MT) Routing in IS-IS*

RFC 5130, *A Policy Control Mechanism in IS-IS Using Administrative Tags*

RFC 5301, *Dynamic Hostname Exchange Mechanism for IS-IS*

RFC 5302, *Domain-wide Prefix Distribution with Two-Level IS-IS*

RFC 5303, *Three-Way Handshake for IS-IS Point-to-Point Adjacencies*

RFC 5304, *IS-IS Cryptographic Authentication*

RFC 5305, *IS-IS Extensions for Traffic Engineering TE*

RFC 5306, *Restart Signaling for IS-IS* – helper mode

RFC 5308, *Routing IPv6 with IS-IS*

RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*

RFC 5310, *IS-IS Generic Cryptographic Authentication*

RFC 6119, *IPv6 Traffic Engineering in IS-IS*

RFC 6213, *IS-IS BFD-Enabled TLV*

RFC 6232, *Purge Originator Identification TLV for IS-IS*

RFC 6233, *IS-IS Registry Extension for Purges*

RFC 6329, *IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging*

RFC 7775, *IS-IS Route Preference for Extended IP and IPv6 Reachability*

RFC 7794, *IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability* – sections 2.1 and 2.3

RFC 7981, *IS-IS Extensions for Advertising Router Information*

RFC 7987, *IS-IS Minimum Remaining Lifetime*

RFC 8202, *IS-IS Multi-Instance* – single topology

RFC 8570, *IS-IS Traffic Engineering (TE) Metric Extensions* – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE

RFC 8919, *IS-IS Application-Specific Link Attributes*

## 7.12 Internet Protocol (IP) Fast Reroute (FRR)

draft-ietf-rtgwg-lfa-manageability-08, *Operational management of Loop Free Alternates*

RFC 5286, *Basic Specification for IP Fast Reroute: Loop-Free Alternates*

RFC 7431, *Multicast-Only Fast Reroute*

RFC 7490, *Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)*

RFC 8518, *Selection of Loop-Free Alternates for Multi-Homed Prefixes*

## 7.13 Internet Protocol (IP) general

draft-grant-tacacs-02, *The TACACS+ Protocol*

RFC 768, *User Datagram Protocol*

RFC 793, *Transmission Control Protocol*

RFC 854, *Telnet Protocol Specifications*

RFC 1350, *The TFTP Protocol (revision 2)*

RFC 2347, *TFTP Option Extension*

RFC 2348, *TFTP Blocksize Option*

RFC 2349, *TFTP Timeout Interval and Transfer Size Options*

RFC 2428, *FTP Extensions for IPv6 and NATs*

RFC 2617, *HTTP Authentication: Basic and Digest Access Authentication*

RFC 2784, *Generic Routing Encapsulation (GRE)*

RFC 2818, *HTTP Over TLS*

RFC 2890, *Key and Sequence Number Extensions to GRE*

RFC 3164, *The BSD syslog Protocol*

RFC 4250, *The Secure Shell (SSH) Protocol Assigned Numbers*

RFC 4251, *The Secure Shell (SSH) Protocol Architecture*

RFC 4252, *The Secure Shell (SSH) Authentication Protocol* – publickey, password

RFC 4253, *The Secure Shell (SSH) Transport Layer Protocol*

RFC 4254, *The Secure Shell (SSH) Connection Protocol*

RFC 4511, *Lightweight Directory Access Protocol (LDAP): The Protocol*

RFC 4513, *Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms* – TLS

RFC 4632, *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*

RFC 5082, *The Generalized TTL Security Mechanism (GTSM)*

RFC 5246, *The Transport Layer Security (TLS) Protocol Version 1.2* – TLS client, RSA public key

RFC 5425, *Transport Layer Security (TLS) Transport Mapping for Syslog* – RFC 3164 with TLS

RFC 5656, *Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer* – ECDSA

RFC 5925, *The TCP Authentication Option*

RFC 5926, *Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)*

RFC 6398, *IP Router Alert Considerations and Usage* – MLD

RFC 6528, *Defending against Sequence Number Attacks*

RFC 7011, *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information*

RFC 7012, *Information Model for IP Flow Information Export*

RFC 7230, *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*

RFC 7231, *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*

RFC 7232, *Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests*

RFC 7301, *Transport Layer Security (TLS) Application Layer Protocol Negotiation Extension*

RFC 7616, *HTTP Digest Access Authentication*

RFC 8446, *The Transport Layer Security (TLS) Protocol Version 1.3*

## 7.14 Internet Protocol (IP) multicast

cisco-ipmulticast/pim-autorp-spec01, *Auto-RP: Automatic discovery of Group-to-RP mappings for IP multicast* – version 1

draft-ietf-bier-pim-signaling-08, *PIM Signaling Through BIER Core*

draft-ietf-idmr-traceroute-ipm-07, *A "traceroute" facility for IP Multicast*

draft-ietf-l2vpn-vpls-pim-snooping-07, *Protocol Independent Multicast (PIM) over Virtual Private LAN Service (VPLS)*

RFC 1112, *Host Extensions for IP Multicasting*

RFC 2236, *Internet Group Management Protocol, Version 2*

RFC 2365, *Administratively Scoped IP Multicast*

RFC 2375, *IPv6 Multicast Address Assignments*

RFC 2710, *Multicast Listener Discovery (MLD) for IPv6*

RFC 3306, *Unicast-Prefix-based IPv6 Multicast Addresses*

RFC 3376, *Internet Group Management Protocol, Version 3*

RFC 3446, *Anycast Rendevous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)*

RFC 3590, *Source Address Selection for the Multicast Listener Discovery (MLD) Protocol*

RFC 3618, *Multicast Source Discovery Protocol (MSDP)*

RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*

RFC 3956, *Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*

RFC 3973, *Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)* – auto-RP groups

RFC 4541, *Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches*

RFC 4604, *Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast*

RFC 4607, *Source-Specific Multicast for IP*

RFC 4608, *Source-Specific Protocol Independent Multicast in 232/8*

RFC 4610, *Anycast-RP Using Protocol Independent Multicast (PIM)*

RFC 4611, *Multicast Source Discovery Protocol (MSDP) Deployment Scenarios*

RFC 5059, *Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*

RFC 5186, *Internet Group Management Protocol Version 3 (IGMPv3) / Multicast Listener Discovery Version 2 (MLDv2) and Multicast Routing Protocol Interaction*

RFC 5384, *The Protocol Independent Multicast (PIM) Join Attribute Format*

RFC 5496, *The Reverse Path Forwarding (RPF) Vector TLV*

RFC 6037, *Cisco Systems' Solution for Multicast in MPLS/BGP IP VPNs*

RFC 6512, *Using Multipoint LDP When the Backbone Has No Route to the Root*

RFC 6513, *Multicast in MPLS/BGP IP VPNs*

RFC 6514, *BGP Encodings and Procedures for Multicast in MPLS/IP VPNs*

RFC 6515, *IPv4 and IPv6 Infrastructure Addresses in BGP Updates for Multicast VPNs*

RFC 6516, *IPv6 Multicast VPN (MVPN) Support Using PIM Control Plane and Selective Provider Multicast Service Interface (S-PMSI) Join Messages*

RFC 6625, *Wildcards in Multicast VPN Auto-Discover Routes*

RFC 6826, *Multipoint LDP In-Band Signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Path*

RFC 7246, *Multipoint Label Distribution Protocol In-Band Signaling in a Virtual Routing and Forwarding (VRF) Table Context*

RFC 7385, *IANA Registry for P-Multicast Service Interface (PMSI) Tunnel Type Code Points*

RFC 7716, *Global Table Multicast with BGP Multicast VPN (BGP-MVPN) Procedures*

RFC 7761, *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*

RFC 8279, *Multicast Using Bit Index Explicit Replication (BIER)*

RFC 8296, *Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks* – MPLS encapsulation

RFC 8401, *Bit Index Explicit Replication (BIER) Support via IS-IS*

RFC 8444, *OSPFv2 Extensions for Bit Index Explicit Replication (BIER)*

RFC 8487, *Mtrace Version 2: Traceroute Facility for IP Multicast*

RFC 8534, *Explicit Tracking with Wildcard Routes in Multicast VPN* – (C-*,C-*) wildcard

RFC 8556, *Multicast VPN Using Bit Index Explicit Replication (BIER)*

## 7.15 Internet Protocol (IP) version 4

RFC 791, *Internet Protocol*

RFC 792, *Internet Control Message Protocol*

RFC 826, *An Ethernet Address Resolution Protocol*

RFC 951, *Bootstrap Protocol (BOOTP)* – relay

RFC 1034, *Domain Names - Concepts and Facilities*

RFC 1035, *Domain Names - Implementation and Specification*

RFC 1191, *Path MTU Discovery* – router specification

RFC 1519, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*

RFC 1534, *Interoperation between DHCP and BOOTP*

RFC 1542, *Clarifications and Extensions for the Bootstrap Protocol*

RFC 1812, *Requirements for IPv4 Routers*

RFC 1918, *Address Allocation for Private Internets*

RFC 2003, *IP Encapsulation within IP*

RFC 2131, *Dynamic Host Configuration Protocol*

RFC 2132, *DHCP Options and BOOTP Vendor Extensions*

RFC 2401, *Security Architecture for Internet Protocol*

RFC 3021, *Using 31-Bit Prefixes on IPv4 Point-to-Point Links*

RFC 3046, *DHCP Relay Agent Information Option (Option 82)*

RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*

RFC 4884, *Extended ICMP to Support Multi-Part Messages* – ICMPv4 and ICMPv6 Time Exceeded

## 7.16 Internet Protocol (IP) version 6

RFC 2464, *Transmission of IPv6 Packets over Ethernet Networks*

RFC 2529, *Transmission of IPv6 over IPv4 Domains without Explicit Tunnels*

RFC 3122, *Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification*

RFC 3315, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*

RFC 3587, *IPv6 Global Unicast Address Format*

RFC 3596, *DNS Extensions to Support IP version 6*

RFC 3633, *IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6*

RFC 3646, *DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*

RFC 3736, *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6*

RFC 3971, *SEcure Neighbor Discovery (SEND)*

RFC 3972, *Cryptographically Generated Addresses (CGA)*

RFC 4007, *IPv6 Scoped Address Architecture*

RFC 4191, *Default Router Preferences and More-Specific Routes* – Default Router Preference

RFC 4193, *Unique Local IPv6 Unicast Addresses*

RFC 4291, *Internet Protocol Version 6 (IPv6) Addressing Architecture*

RFC 4443, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*

RFC 4861, *Neighbor Discovery for IP version 6 (IPv6)*

RFC 4862, *IPv6 Stateless Address Autoconfiguration* – router functions

RFC 4890, *Recommendations for Filtering ICMPv6 Messages in Firewalls*

RFC 4941, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*

RFC 5007, *DHCPv6 Leasequery*

RFC 5095, *Deprecation of Type 0 Routing Headers in IPv6*

RFC 5722, *Handling of Overlapping IPv6 Fragments*

RFC 5798, *Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6* – IPv6

RFC 5952, *A Recommendation for IPv6 Address Text Representation*

RFC 6092, *Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service* – Internet Control and Management, Upper-Layer Transport Protocols, UDP Filters, IPsec and Internet Key Exchange (IKE), TCP Filters

RFC 6106, *IPv6 Router Advertisement Options for DNS Configuration*

RFC 6164, *Using 127-Bit IPv6 Prefixes on Inter-Router Links*

RFC 6221, *Lightweight DHCPv6 Relay Agent*

RFC 6437, *IPv6 Flow Label Specification*

RFC 6603, *Prefix Exclude Option for DHCPv6-based Prefix Delegation*

RFC 8021, *Generation of IPv6 Atomic Fragments Considered Harmful*

RFC 8200, *Internet Protocol, Version 6 (IPv6) Specification*

RFC 8201, *Path MTU Discovery for IP version 6*

## 7.17 Internet Protocol Security (IPsec)

draft-ietf-ipsec-isakmp-mode-cfg-05, *The ISAKMP Configuration Method*

draft-ietf-ipsec-isakmp-xauth-06, *Extended Authentication within ISAKMP/Oakley (XAUTH)*

RFC 2401, *Security Architecture for the Internet Protocol*

RFC 2403, *The Use of HMAC-MD5-96 within ESP and AH*

RFC 2404, *The Use of HMAC-SHA-1-96 within ESP and AH*

RFC 2405, *The ESP DES-CBC Cipher Algorithm With Explicit IV*

RFC 2406, *IP Encapsulating Security Payload (ESP)*

RFC 2407, *IPsec Domain of Interpretation for ISAKMP (IPsec DoI)*

RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)*

RFC 2409, *The Internet Key Exchange (IKE)*

RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec*

RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*

RFC 3526, *More Modular Exponential (MODP) Diffie-Hellman group for Internet Key Exchange (IKE)*

RFC 3566, *The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec*

RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*

RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*

RFC 3947, *Negotiation of NAT-Traversal in the IKE*

RFC 3948, *UDP Encapsulation of IPsec ESP Packets*

RFC 4106, *The Use of Galois/Counter Mode (GCM) in IPsec ESP*

RFC 4109, *Algorithms for Internet Key Exchange version 1 (IKEv1)*

RFC 4301, *Security Architecture for the Internet Protocol*

RFC 4303, *IP Encapsulating Security Payload*

RFC 4307, *Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)*

RFC 4308, *Cryptographic Suites for IPsec*

RFC 4434, *The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)*

RFC 4543, *The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH*

RFC 4754, *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*

RFC 4835, *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*

RFC 4868, *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*

RFC 4945, *The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2 and PKIX*

RFC 5019, *The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments*

RFC 5282, *Using Authenticated Encryption Algorithms with the Encrypted Payload of the IKEv2 Protocol*

RFC 5903, *ECP Groups for IKE and IKEv2*

RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)*

RFC 5998, *An Extension for EAP-Only Authentication in IKEv2*

RFC 6379, *Suite B Cryptographic Suites for IPsec*

RFC 6380, *Suite B Profile for Internet Protocol Security (IPsec)*

RFC 6960, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*

RFC 7296, *Internet Key Exchange Protocol Version 2 (IKEv2)*

RFC 7321, *Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)*

RFC 7383, *Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation*

RFC 7427, *Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)*

## 7.18 Label Distribution Protocol (LDP)

draft-pdutta-mpls-ldp-adj-capability-00, *LDP Adjacency Capabilities*

draft-pdutta-mpls-ldp-v2-00, *LDP Version 2*

draft-pdutta-mpls-mldp-up-redundancy-00, *Upstream LSR Redundancy for Multi-point LDP Tunnels*

draft-pdutta-mpls-multi-ldp-instance-00, *Multiple LDP Instances*

draft-pdutta-mpls-tldp-hello-reduce-04, *Targeted LDP Hello Reduction*

RFC 3037, *LDP Applicability*

RFC 3478, *Graceful Restart Mechanism for Label Distribution Protocol* – helper mode

RFC 5036, *LDP Specification*

RFC 5283, *LDP Extension for Inter-Area Label Switched Paths (LSPs)*

RFC 5443, *LDP IGP Synchronization*

RFC 5561, *LDP Capabilities*

RFC 5919, *Signaling LDP Label Advertisement Completion*

RFC 6388, *Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*

RFC 6512, *Using Multipoint LDP When the Backbone Has No Route to the Root*

RFC 6826, *Multipoint LDP in-band signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*

RFC 7032, *LDP Downstream-on-Demand in Seamless MPLS*

RFC 7473, *Controlling State Advertisements of Non-negotiated LDP Applications*

RFC 7552, *Updates to LDP for IPv6*

## 7.19 Layer Two Tunneling Protocol (L2TP) Network Server (LNS)

draft-mammoliti-l2tp-accessline-avp-04, *Layer 2 Tunneling Protocol (L2TP) Access Line Information Attribute Value Pair (AVP) Extensions*

RFC 2661, *Layer Two Tunneling Protocol "L2TP"*

RFC 2809, *Implementation of L2TP Compulsory Tunneling via RADIUS*

RFC 3438, *Layer Two Tunneling Protocol (L2TP) Internet Assigned Numbers: Internet Assigned Numbers Authority (IANA) Considerations Update*

RFC 3931, *Layer Two Tunneling Protocol - Version 3 (L2TPv3)*

RFC 4719, *Transport of Ethernet Frames over Layer 2 Tunneling Protocol Version 3 (L2TPv3)*

RFC 4951, *Fail Over Extensions for Layer 2 Tunneling Protocol (L2TP) "failover"*

## 7.20 Multiprotocol Label Switching (MPLS)

draft-ietf-mpls-lsp-ping-ospfv3-codepoint-02, *OSPFv3 CodePoint for MPLS LSP Ping*

RFC 3031, *Multiprotocol Label Switching Architecture*

RFC 3032, *MPLS Label Stack Encoding*

RFC 3270, *Multi-Protocol Label Switching (MPLS) Support of Differentiated Services* – E-LSP

RFC 3443, *Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks*

RFC 4023, *Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)*

RFC 4182, *Removing a Restriction on the use of MPLS Explicit NULL*

RFC 4950, *ICMP Extensions for Multiprotocol Label Switching*

RFC 5332, *MPLS Multicast Encapsulations*

RFC 5884, *Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)*

RFC 6374, *Packet Loss and Delay Measurement for MPLS Networks* – Delay Measurement, Channel Type 0x000C

RFC 6424, *Mechanism for Performing Label Switched Path Ping (LSP Ping) over MPLS Tunnels*

RFC 6425, *Detecting Data Plane Failures in Point-to-Multipoint Multiprotocol Label Switching (MPLS) - Extensions to LSP Ping*

RFC 6790, *The Use of Entropy Labels in MPLS Forwarding*

RFC 7308, *Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)*

RFC 7510, *Encapsulating MPLS in UDP*

RFC 7746, *Label Switched Path (LSP) Self-Ping*

RFC 7876, *UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks* – Delay Measurement

RFC 8029, *Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures*

## 7.21 Multiprotocol Label Switching - Transport Profile (MPLS-TP)

RFC 5586, *MPLS Generic Associated Channel*

RFC 5921, *A Framework for MPLS in Transport Networks*

RFC 5960, *MPLS Transport Profile Data Plane Architecture*

RFC 6370, *MPLS Transport Profile (MPLS-TP) Identifiers*

RFC 6378, *MPLS Transport Profile (MPLS-TP) Linear Protection*

RFC 6426, *MPLS On-Demand Connectivity and Route Tracing*

RFC 6427, *MPLS Fault Management Operations, Administration, and Maintenance (OAM)*

RFC 6428, *Proactive Connectivity Verification, Continuity Check and Remote Defect indication for MPLS Transport Profile*

RFC 6478, *Pseudowire Status for Static Pseudowires*

RFC 7213, *MPLS Transport Profile (MPLS-TP) Next-Hop Ethernet Addressing*

## 7.22 Network Address Translation (NAT)

draft-ietf-behave-address-format-10, *IPv6 Addressing of IPv4/IPv6 Translators*

draft-ietf-behave-v6v4-xlate-23, *IP/ICMP Translation Algorithm*

draft-miles-behave-l2nat-00, *Layer2-Aware NAT*

draft-nishitani-cgn-02, *Common Functions of Large Scale NAT (LSN)*

RFC 4787, *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*

RFC 5382, *NAT Behavioral Requirements for TCP*

RFC 5508, *NAT Behavioral Requirements for ICMP*

RFC 6146, *Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers*

RFC 6333, *Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion*

RFC 6334, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite*

RFC 6887, *Port Control Protocol (PCP)*

RFC 6888, *Common Requirements For Carrier-Grade NATs (CGNs)*

RFC 7753, *Port Control Protocol (PCP) Extension for Port-Set Allocation*

RFC 7915, *IP/ICMP Translation Algorithm*

## 7.23 Network Configuration Protocol (NETCONF)

RFC 5277, *NETCONF Event Notifications*

RFC 6020, *YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)*

RFC 6022, *YANG Module for NETCONF Monitoring*

RFC 6241, *Network Configuration Protocol (NETCONF)*

RFC 6242, *Using the NETCONF Protocol over Secure Shell (SSH)*

RFC 6243, *With-defaults Capability for NETCONF*

RFC 8342, *Network Management Datastore Architecture (NMDA)* – Startup, Candidate, Running and Intended datastores

RFC 8525, *YANG Library*

RFC 8526, *NETCONF Extensions to Support the Network Management Datastore Architecture* – <get-data> operation

## 7.24 Open Shortest Path First (OSPF)

RFC 1765, *OSPF Database Overflow*

RFC 2328, *OSPF Version 2*

RFC 3101, *The OSPF Not-So-Stubby Area (NSSA) Option*

RFC 3509, *Alternative Implementations of OSPF Area Border Routers*

RFC 3623, *Graceful OSPF Restart Graceful OSPF Restart* – helper mode

RFC 3630, *Traffic Engineering (TE) Extensions to OSPF Version 2*

RFC 4222, *Prioritized Treatment of Specific OSPF Version 2 Packets and Congestion Avoidance*

RFC 4552, *Authentication/Confidentiality for OSPFv3*

RFC 4576, *Using a Link State Advertisement (LSA) Options Bit to Prevent Looping in BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 4577, *OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 5185, *OSPF Multi-Area Adjacency*

RFC 5187, *OSPFv3 Graceful Restart* – helper mode

RFC 5243, *OSPF Database Exchange Summary List Optimization*

RFC 5250, *The OSPF Opaque LSA Option*

RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*

RFC 5340, *OSPF for IPv6*

RFC 5642, *Dynamic Hostname Exchange Mechanism for OSPF*

RFC 5709, *OSPFv2 HMAC-SHA Cryptographic Authentication*

RFC 5838, *Support of Address Families in OSPFv3*

RFC 6549, *OSPFv2 Multi-Instance Extensions*

RFC 6987, *OSPF Stub Router Advertisement*

RFC 7471, *OSPF Traffic Engineering (TE) Metric Extensions* – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE

RFC 7684, *OSPFv2 Prefix/Link Attribute Advertisement*

RFC 7770, *Extensions to OSPF for Advertising Optional Router Capabilities*

RFC 8362, *OSPFv3 Link State Advertisement (LSA) Extensibility*

RFC 8920, *OSPF Application-Specific Link Attributes*

## 7.25 OpenFlow

TS-007 Version 1.3.1, *OpenFlow Switch Specification* – OpenFlow-hybrid switches

## 7.26 Path Computation Element Protocol (PCEP)

draft-alvarez-pce-path-profiles-04, *PCE Path Profiles*

draft-dhs-spring-pce-sr-p2mp-policy-00, *PCEP extensions for p2mp sr policy*

draft-ietf-pce-binding-label-sid-15, *Carrying Binding Label/Segment Identifier (SID) in PCE-based Networks.* – MPLS binding SIDs

draft-ietf-pce-pceps-tls13-04, *Updates for PCEPS: TLS Connection Establishment Restrictions*

RFC 5440, *Path Computation Element (PCE) Communication Protocol (PCEP)*

RFC 8231, *Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE*

RFC 8253, *PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)*

RFC 8281, *PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model*

RFC 8408, *Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages*

RFC 8664, *Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing*

## 7.27 Point-to-Point Protocol (PPP)

RFC 1332, *The PPP Internet Protocol Control Protocol (IPCP)*

RFC 1990, *The PPP Multilink Protocol (MP)*

RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*

RFC 2516, *A Method for Transmitting PPP Over Ethernet (PPPoE)*

RFC 4638, *Accommodating a Maximum Transit Unit/Maximum Receive Unit (MTU/MRU) Greater Than 1492 in the Point-to-Point Protocol over Ethernet (PPPoE)*

RFC 5072, *IP Version 6 over PPP*

## 7.28 Policy management and credit control

3GPP TS 29.212 Release 11, *Policy and Charging Control (PCC); Reference points* – Gx support as it applies to wireline environment (BNG)

RFC 4006, *Diameter Credit-Control Application*

RFC 6733, *Diameter Base Protocol*

## 7.29 Pseudowire (PW)

draft-ietf-l2vpn-vpws-iw-oam-04, *OAM Procedures for VPWS Interworking*

MFA Forum 12.0.0, *Multiservice Interworking - Ethernet over MPLS*

MFA Forum 13.0.0, *Fault Management for Multiservice Interworking v1.0*

MFA Forum 16.0.0, *Multiservice Interworking - IP over MPLS*

RFC 3916, *Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)*

RFC 3985, *Pseudo Wire Emulation Edge-to-Edge (PWE3)*

RFC 4385, *Pseudo Wire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN*

RFC 4446, *IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)*

RFC 4447, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*

RFC 4448, *Encapsulation Methods for Transport of Ethernet over MPLS Networks*

RFC 5085, *Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires*

RFC 5659, *An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge*

RFC 5885, *Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)*

RFC 6073, *Segmented Pseudowire*

RFC 6310, *Pseudowire (PW) Operations, Administration, and Maintenance (OAM) Message Mapping*

RFC 6391, *Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network*

RFC 6575, *Address Resolution Protocol (ARP) Mediation for IP Interworking of Layer 2 VPNs*

RFC 6718, *Pseudowire Redundancy*

RFC 6829, *Label Switched Path (LSP) Ping for Pseudowire Forwarding Equivalence Classes (FECs) Advertised over IPv6*

RFC 6870, *Pseudowire Preferential Forwarding Status bit*

RFC 7023, *MPLS and Ethernet Operations, Administration, and Maintenance (OAM) Interworking*

RFC 7267, *Dynamic Placement of Multi-Segment Pseudowires*

RFC 7392, *Explicit Path Routing for Dynamic Multi-Segment Pseudowires* – ER-TLV and ER-HOP IPv4 Prefix

RFC 8395, *Extensions to BGP-Signaled Pseudowires to Support Flow-Aware Transport Labels*

## 7.30 Quality of Service (QoS)

RFC 2430, *A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)*

RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*

RFC 2597, *Assured Forwarding PHB Group*

RFC 3140, *Per Hop Behavior Identification Codes*

RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*

## 7.31 Remote Authentication Dial In User Service (RADIUS)

draft-oscca-cfrg-sm3-02, *The SM3 Cryptographic Hash Function*

RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*

RFC 2866, *RADIUS Accounting*

RFC 2867, *RADIUS Accounting Modifications for Tunnel Protocol Support*

RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*

RFC 2869, *RADIUS Extensions*

RFC 3162, *RADIUS and IPv6*

RFC 4818, *RADIUS Delegated-IPv6-Prefix Attribute*

RFC 5176, *Dynamic Authorization Extensions to RADIUS*

RFC 6613, *RADIUS over TCP* – with TLS

RFC 6614, *Transport Layer Security (TLS) Encryption for RADIUS*

RFC 6929, *Remote Authentication Dial-In User Service (RADIUS) Protocol Extensions*

RFC 6911, *RADIUS attributes for IPv6 Access Networks*

## 7.32 Resource Reservation Protocol - Traffic Engineering (RSVP-TE)

draft-newton-mpls-te-dynamic-overbooking-00, *A Diffserv-TE Implementation Model to dynamically change booking factors during failure events*

RFC 2702, *Requirements for Traffic Engineering over MPLS*

RFC 2747, *RSVP Cryptographic Authentication*

RFC 2961, *RSVP Refresh Overhead Reduction Extensions*

RFC 3097, *RSVP Cryptographic Authentication -- Updated Message Type Value*

RFC 3209, *RSVP-TE: Extensions to RSVP for LSP Tunnels*

RFC 3477, *Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)*

RFC 3564, *Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering*

RFC 3906, *Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels*

RFC 4090, *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*

RFC 4124, *Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering*

RFC 4125, *Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*

RFC 4127, *Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*

RFC 4561, *Definition of a Record Route Object (RRO) Node-Id Sub-Object*

RFC 4875, *Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)*

RFC 5712, *MPLS Traffic Engineering Soft Preemption*

RFC 5817, *Graceful Shutdown in MPLS and Generalized MPLS Traffic Engineering Networks*

## 7.33 Routing Information Protocol (RIP)

RFC 1058, *Routing Information Protocol*

RFC 2080, *RIPng for IPv6*

RFC 2082, *RIP-2 MD5 Authentication*

RFC 2453, *RIP Version 2*

## 7.34 Segment Routing (SR)

draft-ietf-bess-mvpn-evpn-sr-p2mp-07, *Multicast and Ethernet VPN with Segment Routing P2MP and Ingress Replication* – MVPN

draft-bashandy-rtgwg-segment-routing-uloop-15, *Loop avoidance using Segment Routing*

draft-filsfils-spring-net-pgm-extension-srv6-usid-15, *Network Programming extension: SRv6 uSID instruction*

draft-filsfils-spring-srv6-net-pgm-insertion-08, *SRv6 NET-PGM extension: Insertion*

draft-ietf-idr-bgpls-srv6-ext-14, *BGP Link State Extensions for SRv6*

draft-ietf-idr-segment-routing-te-policy-23, *Advertising Segment Routing Policies in BGP*

draft-ietf-idr-ts-flowspec-srv6-policy-03, *Traffic Steering using BGP FlowSpec with SR Policy*

draft-ietf-pim-p2mp-policy-ping-03, *P2MP Policy Ping*

draft-ietf-pim-sr-p2mp-policy-06, *Segment Routing Point-to-Multipoint Policy* – MPLS

draft-ietf-rtgwg-segment-routing-ti-lfa-11, *Topology Independent Fast Reroute using Segment Routing*

draft-ietf-spring-conflict-resolution-05, *Segment Routing MPLS Conflict Resolution*

draft-ietf-spring-sr-replication-segment-16, *SR Replication segment for Multi-point Service Delivery* – MPLS

draft-ietf-spring-srv6-srh-compression-xx, *Compressed SRv6 Segment List Encoding in SRH*

draft-voyer-6man-extension-header-insertion-10, *Deployments With Insertion of IPv6 Segment Routing Headers*

RFC 8287, *Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes*

RFC 8426, *Recommendations for RSVP-TE and Segment Routing (SR) Label Switched Path (LSP) Coexistence*

RFC 8476, *Signaling Maximum SID Depth (MSD) Using OSPF* – node MSD

RFC 8491, *Signaling Maximum SID Depth (MSD) Using IS-IS* – node MSD

RFC 8660, *Segment Routing with the MPLS Data Plane*

RFC 8661, *Segment Routing MPLS Interworking with LDP*

RFC 8663, *MPLS Segment Routing over IP* – BGP SR with SR-MPLS-over-UDP/IP

RFC 8665, *OSPF Extensions for Segment Routing*

RFC 8666, *OSPFv3 Extensions for Segment Routing*

RFC 8667, *IS-IS Extensions for Segment Routing*

RFC 8669, *Segment Routing Prefix Segment Identifier Extensions for BGP*

RFC 8754, *IPv6 Segment Routing Header (SRH)*

RFC 8814, *Signaling Maximum SID Depth (MSD) Using the Border Gateway Protocol - Link State*

RFC 8986, *Segment Routing over IPv6 (SRv6) Network Programming*

RFC 9085, *Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing*

RFC 9088, *Signaling Entropy Label Capability and Entropy Readable Label Depth Using IS-IS* – advertising ELC

RFC 9089, *Signaling Entropy Label Capability and Entropy Readable Label Depth Using OSPF* – advertising ELC

RFC 9252, *BGP Overlay Services Based on Segment Routing over IPv6 (SRv6)*

RFC 9256, *Segment Routing Policy Architecture*

RFC 9259, *Operations, Administration, and Maintenance (OAM) in Segment Routing over IPv6 (SRv6)*

RFC 9350, *IGP Flexible Algorithm*

RFC 9352, *IS-IS Extensions to Support Segment Routing over the IPv6 Data Plane*

## 7.35 Simple Network Management Protocol (SNMP)

draft-blumenthal-aes-usm-04, *The AES Cipher Algorithm in the SNMP's User-based Security Model* – CFB128-AES-192 and CFB128-AES-256

draft-ietf-isis-wg-mib-06, *Management Information Base for Intermediate System to Intermediate System (IS-IS)*

draft-ietf-mboned-msdp-mib-01, *Multicast Source Discovery protocol MIB*

draft-ietf-mpls-ldp-mib-07, *Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)*

draft-ietf-mpls-lsr-mib-06, *Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base Using SMIv2*

draft-ietf-mpls-te-mib-04, *Multiprotocol Label Switching (MPLS) Traffic Engineering Management Information Base*

draft-ietf-ospf-mib-update-08, *OSPF Version 2 Management Information Base*

draft-ietf-vrrp-unified-mib-06, *Definitions of Managed Objects for the VRRP over IPv4 and IPv6* – IPv6

ESO-CONSORTIUM-MIB revision 200406230000Z, *esoConsortiumMIB*

IANA-ADDRESS-FAMILY-NUMBERS-MIB revision 200203140000Z, *ianaAddressFamilyNumbers*

IANAifType-MIB revision 200505270000Z, *ianaifType*

IANA-RTPROTO-MIB revision 200009260000Z, *ianaRtProtoMIB*

IEEE8021-CFM-MIB revision 200706100000Z, *ieee8021CfmMib*

IEEE8021-PAE-MIB revision 200101160000Z, *ieee8021paeMIB*

IEEE8023-LAG-MIB revision 200006270000Z, *lagMIB*

LLDP-MIB revision 200505060000Z, *lldpMIB*

RFC 1157, *A Simple Network Management Protocol (SNMP)*

RFC 1212, *Concise MIB Definitions*

RFC 1215, *A Convention for Defining Traps for use with the SNMP*

RFC 1724, *RIP Version 2 MIB Extension*

RFC 1901, *Introduction to Community-based SNMPv2*

RFC 2021, *Remote Network Monitoring Management Information Base Version 2 using SMIv2*

RFC 2206, *RSVP Management Information Base using SMIv2*

RFC 2213, *Integrated Services Management Information Base using SMIv2*

RFC 2494, *Definitions of Managed Objects for the DS0 and DS0 Bundle Interface Type*

RFC 2578, *Structure of Management Information Version 2 (SMIv2)*

RFC 2579, *Textual Conventions for SMIv2*

RFC 2580, *Conformance Statements for SMIv2*

RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*

RFC 2819, *Remote Network Monitoring Management Information Base*

RFC 2856, *Textual Conventions for Additional High Capacity Data Types*

RFC 2863, *The Interfaces Group MIB*

RFC 2864, *The Inverted Stack Table Extension to the Interfaces Group MIB*

RFC 2933, *Internet Group Management Protocol MIB*

RFC 3014, *Notification Log MIB*

RFC 3165, *Definitions of Managed Objects for the Delegation of Management Scripts*

RFC 3231, *Definitions of Managed Objects for Scheduling Management Operations*

RFC 3273, *Remote Network Monitoring Management Information Base for High Capacity Networks*

RFC 3410, *Introduction and Applicability Statements for Internet Standard Management Framework*

RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*

RFC 3412, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*

RFC 3413, *Simple Network Management Protocol (SNMP) Applications*

RFC 3414, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*

RFC 3415, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*

RFC 3416, *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*

RFC 3417, *Transport Mappings for the Simple Network Management Protocol (SNMP)* – SNMP over UDP over IPv4

RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*

RFC 3419, *Textual Conventions for Transport Addresses*

RFC 3498, *Definitions of Managed Objects for Synchronous Optical Network (SONET) Linear Automatic Protection Switching (APS) Architectures*

RFC 3584, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*

RFC 3592, *Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type*

RFC 3593, *Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals*

RFC 3635, *Definitions of Managed Objects for the Ethernet-like Interface Types*

RFC 3637, *Definitions of Managed Objects for the Ethernet WAN Interface Sublayer*

RFC 3826, *The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model*

RFC 3877, *Alarm Management Information Base (MIB)*

RFC 3895, *Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Types*

RFC 3896, *Definitions of Managed Objects for the DS3/E3 Interface Type*

RFC 4001, *Textual Conventions for Internet Network Addresses*

RFC 4022, *Management Information Base for the Transmission Control Protocol (TCP)*

RFC 4113, *Management Information Base for the User Datagram Protocol (UDP)*

RFC 4220, *Traffic Engineering Link Management Information Base*

RFC 4273, *Definitions of Managed Objects for BGP-4*

RFC 4292, *IP Forwarding Table MIB*

RFC 4293, *Management Information Base for the Internet Protocol (IP)*

RFC 4631, *Link Management Protocol (LMP) Management Information Base (MIB)*

RFC 4878, *Definitions and Managed Objects for Operations, Administration, and Maintenance (OAM) Functions on Ethernet-Like Interfaces*

RFC 7420, *Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module*

RFC 7630, *HMAC-SHA-2 Authentication Protocols in the User-based Security Model (USM) for SNMPv3*

SFLOW-MIB revision 200309240000Z, *sFlowMIB*

## 7.36 Timing

GR-1244-CORE Issue 3, *Clocks for the Synchronized Network: Common Generic Criteria*

GR-253-CORE Issue 3, *SONET Transport Systems: Common Generic Criteria*

IEEE 1588-2008, *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*

ITU-T G.781, *Synchronization layer functions*

ITU-T G.811, *Timing characteristics of primary reference clocks*

ITU-T G.813, *Timing characteristics of SDH equipment slave clocks (SEC)*

ITU-T G.8261, *Timing and synchronization aspects in packet networks*

ITU-T G.8262, *Timing characteristics of synchronous Ethernet equipment slave clock (EEC)*

ITU-T G.8262.1, *Timing characteristics of an enhanced synchronous Ethernet equipment slave clock (eEEC)*

ITU-T G.8264, *Distribution of timing information through packet networks*

ITU-T G.8265.1, *Precision time protocol telecom profile for frequency synchronization*

ITU-T G.8272 , *Timing characteristics of primary reference time clocks* – PRTC-A, PRTC-B

ITU-T G.8275.1, *Precision time protocol telecom profile for phase/time synchronization with full timing support from the network*

ITU-T G.8275.2, *Precision time protocol telecom profile for phase/time synchronization with partial timing support from the network*

RFC 3339, *Date and Time on the Internet: Timestamps*

RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*

RFC 8573, *Message Authentication Code for the Network Time Protocol*

## 7.37 Two-Way Active Measurement Protocol (TWAMP)

RFC 5357, *A Two-Way Active Measurement Protocol (TWAMP)* – server, unauthenticated mode

RFC 5938, *Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)*

RFC 6038, *Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features*

RFC 8545, *Well-Known Port Assignments for the One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP)* – TWAMP

RFC 8762, *Simple Two-Way Active Measurement Protocol* – unauthenticated

RFC 8972, *Simple Two-Way Active Measurement Protocol Optional Extensions* – unauthenticated

## 7.38 Virtual Private LAN Service (VPLS)

RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*

RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*

RFC 5501, *Requirements for Multicast Support in Virtual Private LAN Services*

RFC 6074, *Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)*

RFC 7041, *Extensions to the Virtual Private LAN Service (VPLS) Provider Edge (PE) Model for Provider Backbone Bridging*

RFC 7117, *Multicast in Virtual Private LAN Service (VPLS)*

## 7.39 Voice and video

DVB BlueBook A86, *Transport of MPEG-2 TS Based DVB Services over IP Based Networks*

ETSI TS 101 329-5 Annex E, *QoS Measurement for VoIP - Method for determining an Equipment Impairment Factor using Passive Monitoring*

ITU-T G.1020 Appendix I, *Performance Parameter Definitions for Quality of Speech and other Voiceband Applications Utilizing IP Networks - Mean Absolute Packet Delay Variation & Markov Models*

ITU-T G.107, *The E Model - A computational model for use in planning*

ITU-T P.564, *Conformance testing for voice over IP transmission quality assessment models*

RFC 3550, *RTP: A Transport Protocol for Real-Time Applications* – Appendix A.8

RFC 4585, *Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)*

RFC 4588, *RTP Retransmission Payload Format*

## 7.40 Yet Another Next Generation (YANG)

RFC 6991, *Common YANG Data Types*

RFC 7950, *The YANG 1.1 Data Modeling Language*

RFC 7951, *JSON Encoding of Data Modeled with YANG*

## 7.41 Yet Another Next Generation (YANG) OpenConfig Models

openconfig-aaa.yang version 0.4.0, *OpenConfig AAA Model*

openconfig-aaa-radius.yang version 0.3.0, *OpenConfig AAA RADIUS Model*

openconfig-aaa-tacacs.yang version 0.3.0, *OpenConfig AAA TACACS+ Model*

openconfig-acl.yang version 1.0.0, *OpenConfig ACL Model*

openconfig-alarms.yang version0.3.2, *OpenConfig System Alarms Model*

openconfig-bfd.yang version 0.2.2, *OpenConfig BFD Model*

openconfig-bgp.yang version 6.1.0, *OpenConfig BGP Model*

openconfig-bgp-common.yang version 6.0.0, *OpenConfig BGP Common Model*

openconfig-bgp-common-multiprotocol.yang version 6.0.0, *OpenConfig BGP Common Multiprotocol Model*

openconfig-bgp-common-structure.yang version 6.0.0, *OpenConfig BGP Common Structure Model*

openconfig-bgp-global.yang version 6.0.0, *OpenConfig BGP Global Model*

openconfig-bgp-neighbor.yang version 6.1.0, *OpenConfig BGP Neighbor Model*

openconfig-bgp-peer-group.yang version 6.1.0, *OpenConfig BGP Peer Group Model*

openconfig-bgp-policy.yang version 4.0.1, *OpenConfig BGP Policy Model*

openconfig-if-aggregate.yang version 2.4.3, *OpenConfig Interfaces Aggregated Model*

openconfig-if-ethernet.yang version 2.12.1, *OpenConfig Interfaces Ethernet Model*

openconfig-if-ip.yang version 3.1.0, *OpenConfig Interfaces IP Model*

openconfig-if-ip-ext.yang version 2.3.1, *OpenConfig Interfaces IP Extensions Model*

openconfig-igmp.yang version 0.2.0, *OpenConfig IGMP Model*

openconfig-interfaces.yang version 3.0.0, *OpenConfig Interfaces Model*

openconfig-isis.yang version 1.1.0, *OpenConfig IS-IS Model*

openconfig-isis-policy.yang version 0.5.0, *OpenConfig IS-IS Policy Model*

openconfig-isis-routing.yang version 1.1.0, *OpenConfig IS-IS Routing Model*

openconfig-lacp.yang version 1.3.0, *OpenConfig LACP Model*

openconfig-lldp.yang version 0.1.0, *OpenConfig LLDP Model*

openconfig-local-routing.yang version 1.2.0, *OpenConfig Local Routing Model*

openconfig-mpls.yang version 2.3.0, *OpenConfig MPLS Model*

openconfig-mpls-ldp.yang version 3.0.2, *OpenConfig MPLS LDP Model*

openconfig-mpls-rsvp.yang version 2.3.0, *OpenConfig MPLS RSVP Model*

openconfig-mpls-te.yang version 2.3.0, *OpenConfig MPLS TE Model*

openconfig-network-instance.yang version 1.1.0, *OpenConfig Network Instance Model*

openconfig-network-instance-l3.yang version 0.11.1, *OpenConfig L3 Network Instance Model* – static routes

openconfig-ospfv2.yang version 0.4.0, *OpenConfig OSPFv2 Model*

openconfig-ospfv2-area.yang version 0.4.0, *OpenConfig OSPFv2 Area Model*

openconfig-ospfv2-area-interface.yang version 0.4.0, *OpenConfig OSPFv2 Area Interface Model*

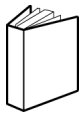
openconfig-ospfv2-common.yang version 0.4.0, *OpenConfig OSPFv2 Common Model*

openconfig-ospfv2-global.yang version 0.4.0, *OpenConfig OSPFv2 Global Model*

openconfig-packet-match.yang version 1.0.0, *OpenConfig Packet Match Model*

openconfig-pim.yang version 0.2.0, *OpenConfig PIM Model*
openconfig-platform.yang version 0.15.0, *OpenConfig Platform Model*
openconfig-platform-fan.yang version 0.1.1, *OpenConfig Platform Fan Model*
openconfig-platform-linecard.yang version 0.1.2, *OpenConfig Platform Linecard Model*
openconfig-platform-port.yang version 0.4.2, *OpenConfig Port Model*
openconfig-platform-transceiver.yang version 0.9.0, *OpenConfig Transceiver Model*
openconfig-procmon.yang version 0.4.0, *OpenConfig Process Monitoring Model*
openconfig-relay-agent.yang version 0.1.0, *OpenConfig Relay Agent Model*
openconfig-routing-policy.yang version 3.0.0, *OpenConfig Routing Policy Model*
openconfig-rsvp-sr-ext.yang version 0.1.0, *OpenConfig RSVP-TE and SR Extensions Model*
openconfig-system.yang version 0.10.1, *OpenConfig System Model*
openconfig-system-grpc.yang version 1.0.0, *OpenConfig System gRPC Model*
openconfig-system-logging.yang version 0.3.1, *OpenConfig System Logging Model*
openconfig-system-terminal.yang version 0.3.0, *OpenConfig System Terminal Model*
openconfig-telemetry.yang version 0.5.0, *OpenConfig Telemetry Model*
openconfig-terminal-device.yang version 1.9.0, *OpenConfig Terminal Optics Device Model*
openconfig-vlan.yang version 2.0.0, *OpenConfig VLAN Model*

# Customer document and product support



## Customer documentation

Customer documentation welcome page

## Technical support

Product support portal

## Documentation feedback

Customer documentation feedback