IP Router Configuration

In This Chapter

This chapter provides information about commands required to configure basic router parameters.

Topics in this chapter include:

Configuring IP Router Parameters

In order to provision services on an Alcatel-Lucent 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 on page 22
- Creating an IP Address Range on page 26
- Autonomous Systems (AS) on page 39
- Confederations on page 40
- Proxy ARP on page 42

Refer to 7750 SROS Triple Play Guide for information about DHCP and support as well as configuration examples. on page 33

Interfaces

Alcatel-Lucent routers use different types of interfaces for various functions. Interfaces must be configured with parameters 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).

Network Interface

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

- A physical or logical port
- A SONET/SDH channel

Network Domains

In order to determine which network ports (and hence which network complexes) are eligible to transport traffic of individual SDPs, network-domain is introduced. This information is then used for the sap-ingress queue allocation algorithm applied to VPLS SAPs. This algorithm is optimized in such a way that no sap-ingress queues are allocated if the given port does not belong to the network-domain used in the given VPLS. In addition, sap-ingress queues will not be allocated towards network ports (regardless of the network-domain membership) if the given VPLS does not contain any SDPs.

Sap-ingress queue allocation takes into account the following aspects:

- SHG membership of individual SDPs
- Network-domain definition under SDP to restrict the topology the given SDP can be set-up in

The implementation supports four network-domains within any given VPLS.

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

Network-domain configuration is irrelevant for Layer 3 services (Layer 3 VPN and/or IES service). It 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 will only be used for ingress VPLS sap queue-allocation. It will not be taken into account by routing during SDP setup. As a consequence, if the given SDP is routed through network interfaces that are not part of the configured network domain, the packets will be still forwarded, but their QoS and queuing behavior will be based on default settings. In addition, the packet will not appear in SAP stats.

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

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

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

System Interface

The system interface is associated with the 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:

- The termination point of service tunnels
- The hops when configuring MPLS paths and LSPs
- The 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 also referred to as the loopback address and is used as the router identifier. A system interface must have an IP address with a 32-bit subnet mask.

Unicast Reverse Path Forwarding Check (uRPF)

This section applies to the 7750-SR, 7710-SR, 7950-SR and the 7450-ESS.
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 IP addresses to allow attackers to thwart efforts to locate or filter the attacks. For Internet service providers (ISPs) that provide public access, Unicast RPF deflects such attacks by forwarding only packets that have 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 packet has a source address with a corresponding 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 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 will pass the uRPF check even when the uRPF mode 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 and even if the interface is not a member of any ECMP routes. The strict-no-ecmp uRPF mode may be configured on any interface which 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 then this is included in the uRPF check, as follows:

If there is a default route:

- A loose mode uRPF check always succeeds.
- A strict mode uRPF check only succeeds if the SA 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 uRPF check.

Creating an IP Address Range

An IP address range can be reserved for exclusive use for services by defining the **config>router>service-prefix** command. When the service is configured, the IP address must be in the range specified as a service prefix. If no service prefix command is configured, then no limitation exists.

Addresses in the range of a service prefix can be allocated to a network port unless the *exclusive* parameter is used. Then, the address range is exclusively reserved for services.

When defining a range that is a superset of a previously defined service prefix, the subset will be replaced with the superset definition. For example, if a service prefix exists for 10.10.10.0/24, and a new service prefix is configured as 10.10.0.0/16, then the old address (10.10.10.0/24) will be replaced with the new address (10.10.0.0/16).

When defining a range that is a subset of a previously defined service prefix, the subset will replace the existing superset, providing addresses used by services are not affected; for example, if a service prefix exists for 10.10.0.0/16, and a new service prefix is configured as 10.10.10.0/24, then the 10.10.0.0/16 entry will be removed, provided that no services are configured that use 10.10.x.x addresses other than 10.10.10.x.

QoS Policy Propagation Using BGP (QPPB)

This section discusses QPPB as it applies to VPRN, IES, and router interfaces. Refer to the Internet Enhanced Service section in the Services Guide and the IP Router Configuration section in the 7x50 SR OS 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. In the industry this feature is called QPPB, and even though the feature name refers to BGP specifically. On SR routers, QPPB is supported for BGP (IPv4, IPv6, VPN-IPv4, VPN-IPv6), RIP and static routes.

While SAP ingress and network QoS policies can achieve the same end result as QPPB, assigning a packet arriving on a particular IP interface to a specific forwarding-class and priority/profile based on the source IP address or destination IP address of the packet 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 particular 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.

QPPB Applications

There are two typical applications of QPPB:

1. Coordination of QoS policies between different administrative domains.
2. Traffic differentiation within a single domain, based on route characteristics.

Inter-AS Coordination of QoS Policies

The operator of an administrative domain A can use QPPB to signal to a peer administrative domain B that traffic sent to certain prefixes advertised by domain A should receive a particular QoS treatment in domain B. More specifically, an ASBR of domain A can advertise a prefix XYZ to domain B and include a BGP community attribute with the route. The community value implies a particular 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 XYZ into their routing table, they apply a QoS policy on selected interfaces that classifies traffic towards network XYZ into the QoS class implied by the BGP community value.

QPPB may also be used to request that traffic sourced from certain 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 discussed above. However, in this case other approaches are equally valid, such as marking the DSCP or other CoS fields based on source IP address so that downstream domains can take action based on a common understanding of the QoS treatment implied by different DSCP values.

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

Traffic Differentiation Based on Route Characteristics

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

Figure 1 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 we presume that 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. Note however, that the DSCP or other COS markings could be left unchanged in the ISP's network and QPPB used on every node.

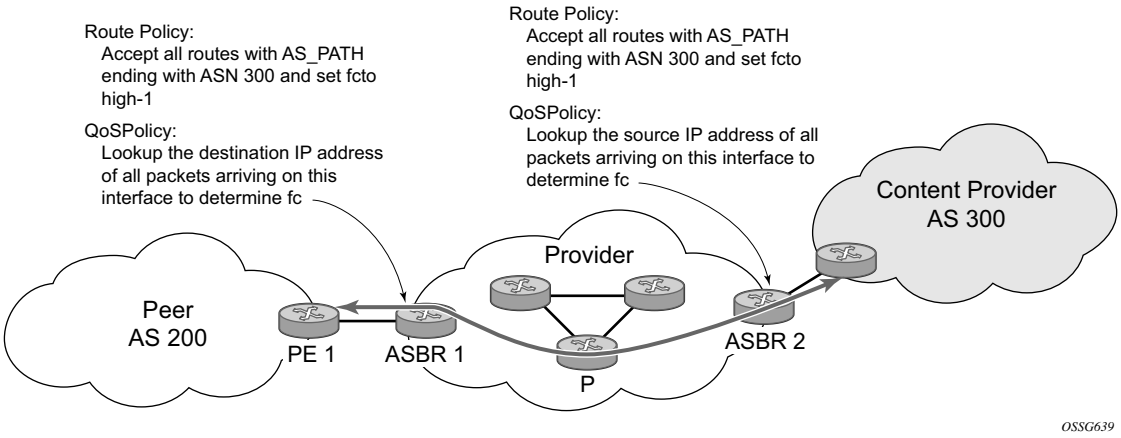

Figure 1: Use of QPPB to Differentiate Traffic in an ISP Network

QPPB

There are two main aspects of the QPPB feature:

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

Associating an FC and Priority with a Route

This feature uses a 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 command has the following structure:

```
fc fc-name [priority {low | high}]
```

The use of this command is illustrated by the following example:

```
config>router>policy-options
     begin
     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
     commit
```

The **fc** command is supported with all existing from and to match conditions in a route policy entry and with any action other than reject, it is supported 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 types of policies:

- VRF import policies:

  → config>service>vprn>vrf-import

- BGP import policies:
  - → config>router>bgp>import
  - → config>router>bgp>group>import
  - → config>router>bgp>group>neighbor>import
  - → config>service>vprn>bgp>import
  - → config>service>vprn>bgp>group>import
  - → config>service>vprn>bgp>group>neighbor>import
- RIP import policies:
  - → config>router>rip>import
  - → config>router>rip>group>import
  - → config>router>rip>group>neighbor>import
  - → config>service>vprn>rip>import
  - → config>service>vprn>rip>group>import
  - → config>service>vprn>rip>group>neighbor>import

As evident from above, 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 address families listed below:

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

Note that 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 and then the BGP import policy, so the QPPB QoS is based on the BGP import policy entry.

This feature also introduces the ability to associate a forwarding-class and optionally priority with IPv4 and IPv6 static routes. This is achieved using the following modified versions of the static-route commands:

- static-route {*ip-prefix*/*prefix-length*|*ip-prefix netmask*} [fc *fc-name* [priority {low | high}]] next-hop *ip-int-name*|*ip-address*
- static-route {*ip-prefix*/*prefix-length*|*ip-prefix netmask*} [fc *fc-name* [priority {low | high}]] indirect *ip-address*

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

Displaying QoS Information Associated with Routes

The following commands are enhanced 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

This feature uses a **qos** keyword to the **show>router>route-table** command. When this 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 fc and priority information then the third line is blank. The following CLI shows an example:

show router route-table [**family**] [*ip-prefix*[/*prefix-length*]] [**longer** | **exact**] [**protocol** *protocol-name*] **qos**

An example output of this command is shown below:

```
A:Dut-A# show router route-table 10.1.5.0/24 qos
===============================================================================
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
===============================================================================
A:Dut-A#
```

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 the **qos-route-lookup** command is necessary in the configuration of the IP interface. The **qos-route-lookup** command has parameters 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 enabled for IPv4 only, IPv6 only, or both IPv4 and IPv6. Note however, current QPPB based on a source IP address is not supported for IPv6 packets nor is it supported 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 (config>router>interface)
- VPRN SAP and spoke SDP interfaces (config>service>vprn>interface)
- VPRN group-interfaces (config>service>vprn>sub-if>grp-if)
- IES SAP and spoke SDP interfaces (config>service>ies>interface)
- IES group-interfaces (config>service>ies>sub-if>grp-if)

When the qos-route-lookup command with the destination parameter 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, overriding the fc and priority/profile determined from the sap-ingress or network qos policy associated with the IP interface (see section 5.7 for further details). 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.

Similarly, when the qos-route-lookup command with the source parameter 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, overriding 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.

Currently, QPPB is not supported for ingress MPLS traffic on network interfaces or on CsC PE'-CE' interfaces (config>service>vprn>nw-if).

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

QPPB When Next-Hops are Resolved by QPPB Routes

In some circumstances (IP VPN inter-AS model C, Carrier Supporting Carrier, indirect static routes, etc.) 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 and perhaps N2 is resolved by a route A3 with next-hop N3, etc. In release 9.0 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 then the QoS classification is not based on QPPB, even if routes A2, A3, etc. have forwarding-class and priority associations.

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 given flow on the same path. The QPPB configuration model described in Associating an FC and Priority with a Route on page 30 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 particular 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 in addition to 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 then the backup next-hop is used. The QPPB configuration model described in Associating an FC and Priority with a Route on page 30 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.

QPPB and Policy-Based Routing

When an IPv4 or IPv6 packet with destination address X 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 X 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 X is matched by a route with a forwarding-class and priority

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.

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 implications listed below:

- 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 then the packet will be 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 then the packet is assigned this profile state. In both cases there is no consideration of whether or not **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 then priority will be 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 then the packet priority will be 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 then the packet priority will be based on QPPB (unless DE=1), but if no priority information is associated with the route then the packet priority will be based on the configuration of **fc1** (if **fc1** mapped to a priority mode queue then it is based on DSCP/IP prec/802.1p and if **fc1** mapped to a profile mode queue then it is based on the profile state of **fc1**).

Table 3 summarizes these interactions.

Table 3: 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 is 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
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

Table 3: QPPB Interactions with SAP Ingress QoS (Continued)

Original FC object mapping	New FC object mapping	Profile	Priority (drop preference)	DE=1 override	In/out of profile marking
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 is 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 is high priority	From new base FC	From original FC and sub-class

Router ID

The router ID, a 32-bit number, uniquely identifies the router within an autonomous system (AS) (see Autonomous Systems (AS) on page 39). 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 derived in the following ways.

- Define the value in the **config>router** *router-id* context. The value becomes the router ID.
- Configure the system interface with an IP address in the **config>router>interface** *ip-int-name* context. If the router ID is not manually configured in the **config>router** *router-id* context, then the system interface acts as the router ID.
- If neither the system interface or router ID are implicitly specified, then the router ID is inherited from the last four bytes of the MAC address.
- The router can be derived on the protocol level; for example, BGP.

Autonomous Systems (AS)

Networks can be grouped into areas. An area is a collection of network segments within an 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 ASs 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.

Confederations

Configuring confederations is optional and should only be implemented to reduce the 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. Parameter values such as next hop, metric, and local preference settings 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 — 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 depicts a confederation configuration example.

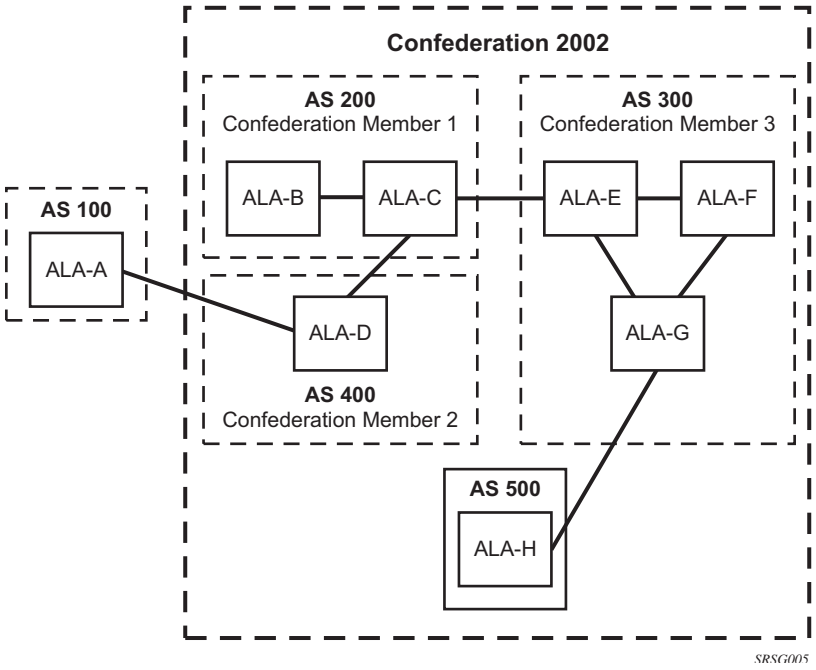

Figure 2: Confederation Configuration

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.

In order 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 will be attempted and prefix lists that determine for which source hosts proxy ARP will be attempted.

In addition, the proxy ARP implementation will support 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 can not reach each other directly.

Static ARP is used when an Alcatel-Lucent router needs to know about a device on an interface that cannot or does not respond to ARP requests. Thus, the configuration can state that if it has a packet with a certain 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.

Exporting an Inactive BGP Route from a VPRN

The **export-inactive-bgp** command under config>service>vprn introduces an IP VPN configuration 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 due to the presence of a more preferred BGP-VPN route from another PE. This “best-external” type of route advertisement is useful in active/standby multi-homing scenarios because it can ensure that all PEs have knowledge of the backup path provided by the standby PE.

DHCP Relay

Refer to 7750 SROS Triple Play Guide for information about DHCP and support provided by the 7750 SR as well as configuration examples.

Internet Protocol Versions

The TiMOS 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 newer version of the Internet Protocol designed as a successor to IP version 4 (IPv4) (RFC-791, *Internet Protocol*). The changes from IPv4 to IPv6 effect 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 new 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 new options in the future.
- Flow labeling capability — The capability to enable the labeling of packets belonging to particular 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.

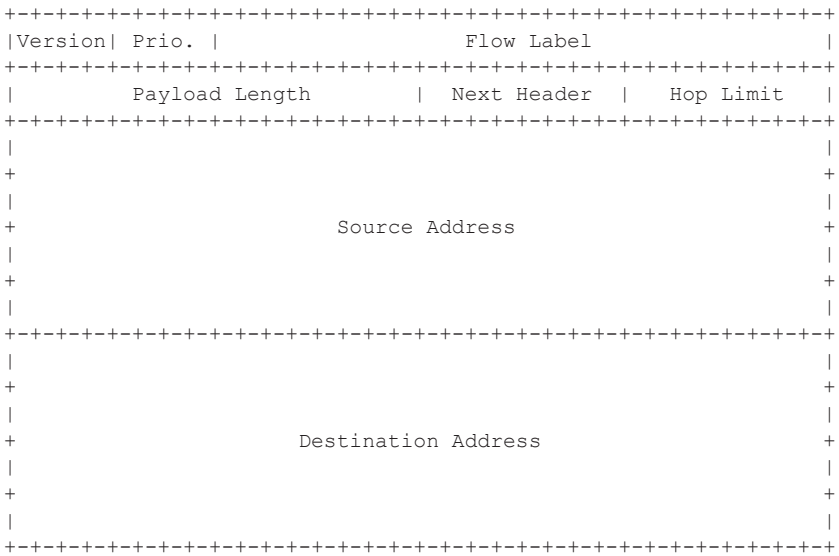

Figure 3: IPv6 Header Format

Table 4: 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).

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 once in an address.

The IPv6 prefix is the part of the IPv6 address that represents the network identifier. The network identifier appears at the beginning of the IP 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 1080:6809:8086:6502::1/64 means that the first 64 bits of the address represent the network identifier; the remaining 64 bits represent the node identifier.

Note: In SR OS 12.0.R4 any function that displays an IPv6 address or prefix changes to reflect rules described in RFC 5952, *A Recommendation for IPv6 Address Text Representation*. Specifically, hexadecimal letters in IPv6 addresses are now represented in lowercase, and the correct compression of all leading zeros is displayed. This changes visible display output compared to previous SR OS releases. Previous SR OS behavior can cause issues with operator scripts that use standard IPv6 address expressions and with libraries that have standard IPv6 parsing as per RFC 5952 rules.

IPv6 Applications

Examples of the IPv6 applications supported by the TiMOS include:

- IPv6 Internet exchange peering — Figure 4 shows an IPv6 Internet exchange where multiple ISPs peer over native IPv6.

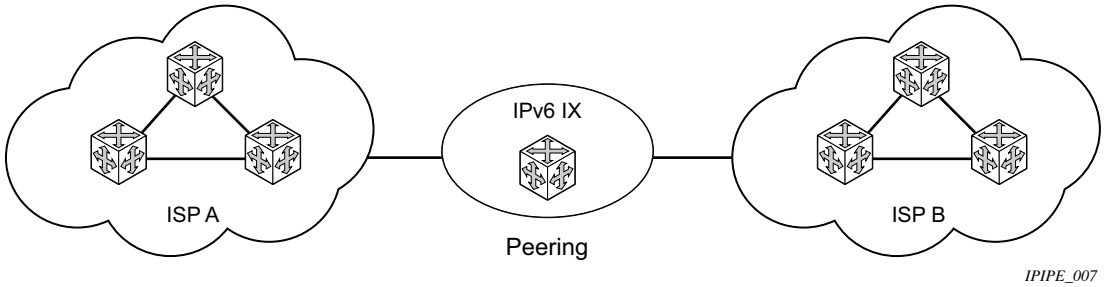

Figure 4: IPv6 Internet Exchange

- IPv6 transit services — Figure 5 shows IPv6 transit provided by an ISP.

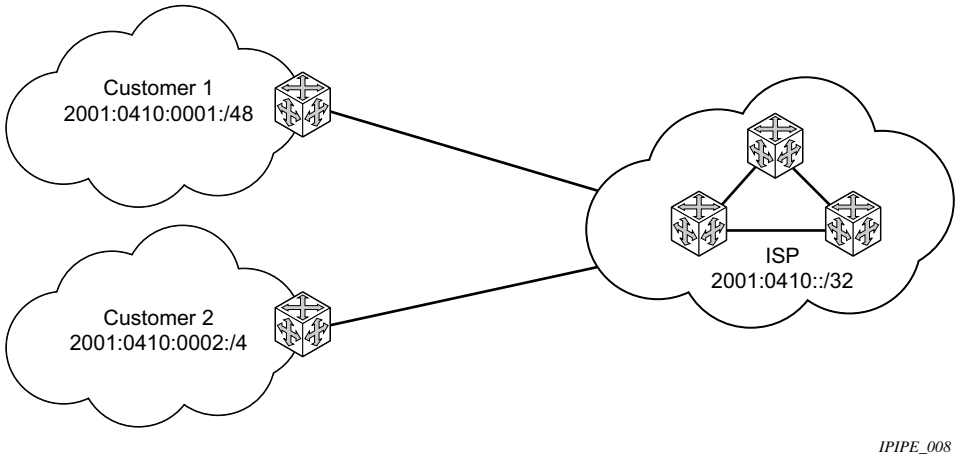

Figure 5: IPv6 Transit Services

- IPv6 services to enterprise customers and home users — Figure 6 shows IPv6 connectivity to enterprise and home broadband users.

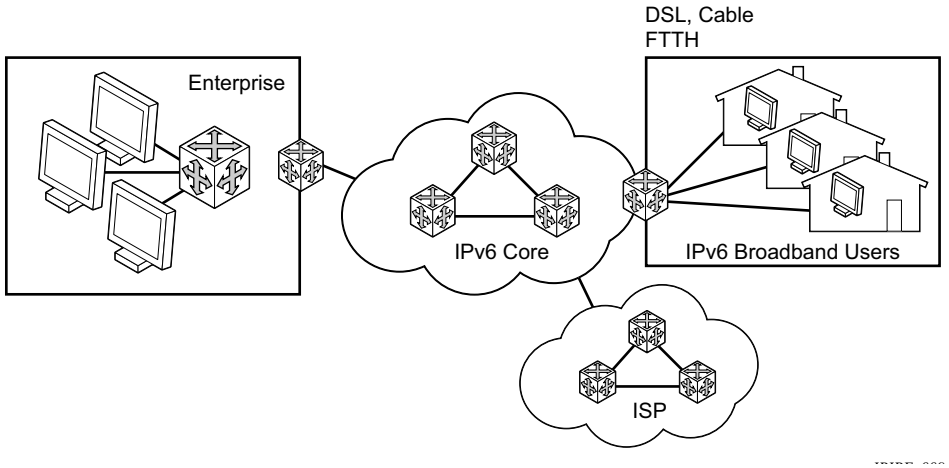

Figure 6: IPv6 Services to Enterprise Customers and Home Users

- 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. Alcatel-Lucent router supports 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 6over4 tunnel.

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

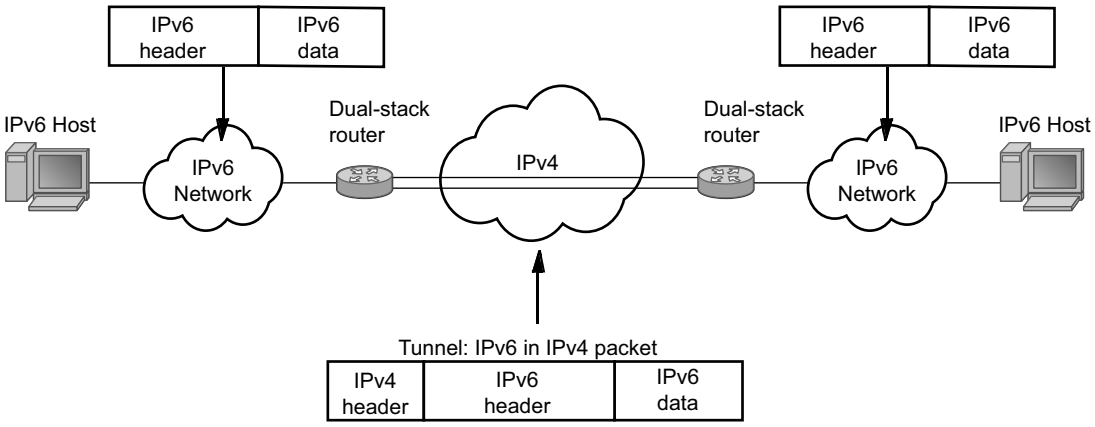

Figure 7: IPv6 over IPv4 Tunnels

DNS

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

Secure Neighbor Discovery (SeND)

Secure Neighbor Discovery (SeND) in conjunction with Cryptographically Generated Addresses (CGAs) introduce a concept that allows operators 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 will generate a CGA from the configured prefix (GUA, LLA) and use that address for all communication. The router will validate NS/ND messages from other nodes on the network segment, and only install them in the neighbor cache if they pass validation.

A number of potential use-cases for SeND exist in order to secure the network from deliberate or accidental tampering during neighbor discovery; principally to 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 at all; or to validate which routers are permitted to act as default gateways.

SeND impacts the following areas of neighbor discovery:

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

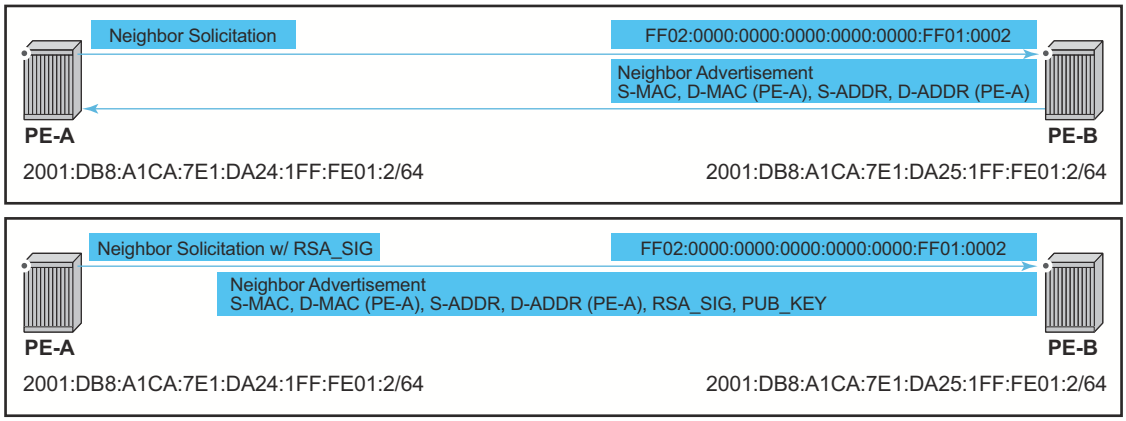

Figure 8: Neighbor discovery with and without SeND

When SeND is enabled on a node, basic neighbor discovery messaging is changed as illustrated in Figure 8. In the example, PE-A wants 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 as it is configured for SeND operation, processes the NS. PE-B will validate 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 a 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, then both nodes will install each other’s addresses into their neighbor cache database.

SeND Persistent CGAs

Persistent CGAs is an enhancement of the SeND feature, introduced in release 12.

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. Make sure that the CompactFlash (CF) file(s) containing an RSA key pair that is used for SeND, is (are) synchronized to the standby CPM by making use of the HA infrastructure used for certificates.
3. Make sure the configuration file is saved when one or more CGAs are generated.

Persistent RSA Key Pair

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

Generate an RSA Key Pair

To generate an RSA key pair, use the **admin certificate gen-keypair** command:

admin certificate gen-keypair <local-url> [type rsa] size 1024

For example

```
admin certificate gen-keypair cf1:\myDir\myRsaKeyPair type rsa size 1024
```

This generates a der formatted file.

Import an online/offline generated RSA key pair

To import a generated RSA key pair, use the **admin certificate secure-nd-import** command:

admin certificate secure-nd-import <local-url> format der|pem|pkcs12 [password <[32 chars max]>] [key-rollover]

For example

```
admin certificate secure-nd-import cf1:\myDir\myRsaKeyPair format der
```

- Since SeND only uses RSA key pairs, the command is refused if the imported key type is not RSA.
- Since 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.
- **key-rollover** keyword: see the *RSA key pair rollover mechanism* section that follows.

- Creates the file cfx:\system-pki\secureNdKey (fixed directory and file name) and saves the imported key in that file in encrypted der format (same as the **admin certificate import** command).
- The RSA key pair is uploaded in the memory of SeND.

RSA key pair rollover mechanism

To trigger a key rollover, use the **admin certificate secure-nd-import** command described in the previous section "Import an online/offline generated RSA key pair".

For example

```
admin certificate secure-nd-import cf1:\myDir\myOtherRsaKeyPair format
der key-rollover
```

- If CGAs exist that are generated based on an auto-generated 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 will be regenerated.
- While handling a key rollover, SeND keeps track of which interface uses which RSA key pair. Hence 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.

Auto-generation of 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 operator did not import an RSA key pair for SeND, an auto-generated RSA key pair will be used as a fallback.

The auto-generated RSA key pair is synced to the standby CPM as it is done in the previous release, but it will not be written to the CF. Therefore, all CGAs generated via an auto-generated RSA key pair, are not persistent. A warning will be given whenever a non-persistent CGA is generated.

The **admin certificate secure-nd-import** command without the **key-rollover** keyword will be refused if CGAs exist that made use of the auto-generated RSA key pair. Specifying the **key-rollover** keyword will result in regeneration of the CGAs.

See the section “Making non-persistent CGAs persistent” for more information on 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, the following commands for automatic and manual certificate synchronization are used:

- manual: **admin redundancy synchronize cert**
- automatic: **configure redundancy cert-sync**

SeND also synchronizes the RSA key pair to the standby CPM as it is done in the previous release.

Persistent CGA Modifier

The modifier used during the CGA generation will be 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.

Note that the modifier is needed to be sent out in ND messages.

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

Example1: Configure a SeND interface without modifiers (as it is done in release 12.0).

```
configure router interface itf1
    address 10.10.10.1
    port 1/1/1
    ipv6
         secure-nd
              no shutdown
```

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

=> The modifier is saved in the interface configuration file.

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

=> 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 2: Configure a SeND interface with modifiers.

```
configure router interface itf2
    address 10.10.10.2
    port 1/1/2
    ipv6
         secure-nd
              link-local-modifier 0xABCD
```

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

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

=> 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.

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

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

```
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 done 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 this case, the modifier will be replaced by a new one that is generated based on the actual RSA key pair.

Making non-persistent CGAs persistent

CGAs can be non-persistent because:

- The operator forgot to configure an RSA key pair for SeND and hence the CGAs were generated based on an auto-generated RSA key pair.
- The operator forgot to synchronize an RSA key pair file to the stand-by CPM and a switch-over 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 will result 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:

admin certificate secure-nd-export

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

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 will be 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 given to the operator that a new CGA is generated.

If a modifier is missing in the configuration file for an IPv6 /64 prefix on a SeND enabled interface, a new modifier and CGA will be 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.

IPv6 Provider Edge Router over MPLS (6PE)

6PE allows IPv6 domains to communicate with each other over an IPv4 MPLS core network. This architecture requires no backbone infrastructure upgrades and no re-configuration of core routers, because forwarding is purely based on MPLS labels. 6PE is a cost effective solution for IPv6 deployment.

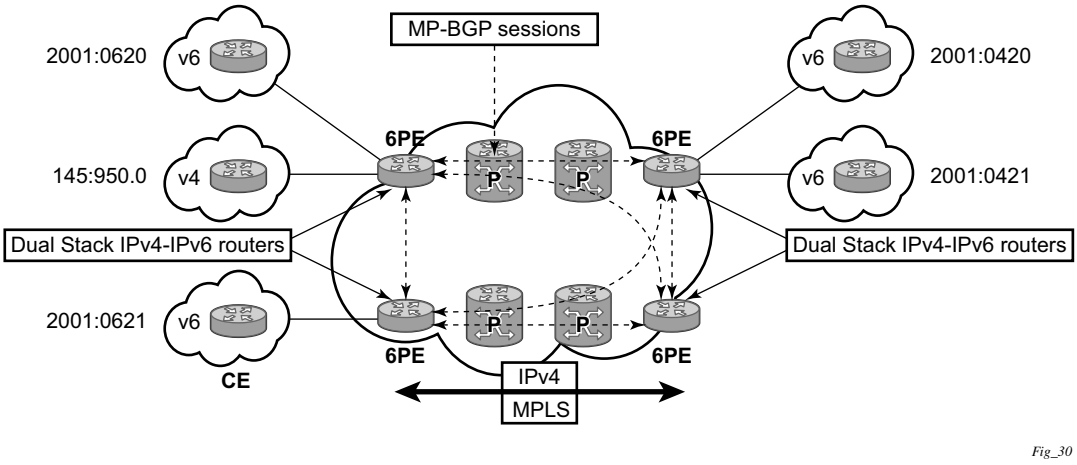

Figure 9: Example of a 6PE Topology within One AS

6PE Control Plane Support

The 6PE MP-BGP routers support:

- IPv4/IPv6 dual-stack
- MP-BGP can be used between 6PE routers to exchange IPv6 reachability information.
  - → The 6PE routers exchange IPv6 prefixes over MP-BGP sessions running over IPv4 transport. The MP-BGP AFI used is IPv6 (value 2).
  - → An IPv4 address of the 6PE router is encoded as an IPv4-mapped IPv6 address in the BGP next-hop field of the IPv6 NLRI. By default, the IPv4 address that is used for peering is used. It is configurable through the route policies.
  - → The 6PE router binds MPLS labels to the IPv6 prefixes it advertises. The SAFI used in MP-BGP is the SAFI (value 4) label. The router uses the IPv6 explicit null (value 2) label for all the IPv6 prefixes that it advertises and can accept an arbitrary label from its peers.
- LDP is used to create the MPLS full mesh between the 6PE routers and the IPv4 addresses that are embedded in the next-hop field are reachable by LDP LSPs. The ingress 6PE router uses the LDP LSPs to reach remote 6PE routers.

6PE Data Plane Support

The ingress 6PE router can push two MPLS labels to send the packets to the egress 6PE router. The top label is an LDP label used to reach the egress 6PE router. The bottom label is advertised in MP-BGP by the remote 6PE router. Typically, the IPv6 explicit null (value 2) label is used but an arbitrary value can be used when the remote 6PE router is from a vendor other than Alcatel-Lucent.

The egress 6PE router pops the top LDP tunnel label. It sees the IPv6 explicit null label, which indicates an IPv6 packet is encapsulated. It also pops the IPv6 explicit null label and performs an IPv6 route lookup to find out the next hop for the IPv6 packet.

Static Route Resolution Using Tunnels

The user can forward packets of a static route to an indirect next-hop over a tunnel programmed in TTM by configuring the following static route tunnel binding command:

```
config>router>static-route-entry {ip-prefix/prefix-length} [mcast] indirect {ip-address}
tunnel-next-hop
     resolution {any|disabled|filter}
     resolution-filter
          [no] ldp
          [no] rsvp-te
               [no] [lsp <name1>]
               [no] [lsp <name2>]
               .
               .
               [no] [lsp <namen>]
          exit
     [no] disallow-igp
     exit
exit
```

The **static-route-entry** command is only supported with the **indirect** next-hop option and the **tunnel-next-hop** option configured together. The existing **static-route** command is still supported with all other options, including the **indirect** option which can be used to resolve the indirect next-hops in RTM.

The new command is an add-on to configure the resolution to tunnel next-hops in TTM. As such, the user must first configure the prefix with the existing command and the **indirect** option and then enter the new **static-route-entry** command with the **indirect** option. For example:

```
/configure router static-route 5.5.5.5/32 indirect 1.0.0.2
/configure router static-route-entry 5.5.5.5/32 indirect 1.0.0.2
     tunnel-next-hop
          rsvp-te
               lsp to-1.0.0.2-1
               lsp to-1.0.0.2-2
          exit
     no shutdown
exit
```

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

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

The following tunnel types are supported in a static route context: RSVP and LDP.

- The **ldp** value instructs the code to search for an LDP LSP with a FEC prefix corresponding to the address of the indirect next-hop.

- The **rsvp** value instructs the code to search for the best metric RSVP LSP to the address of the indirect next-hop. This address can correspond to the system interface or to another loopback used on the remote node. The LSP metric is provided by MPLS in the tunnel table. In the case of multiple RSVP LSPs with the same lowest metric, the code selects the LSP with the lowest tunnel-id.

If one or more explicit tunnel types are specified using the **resolution-filter** option, then only these tunnel types will be selected again following the TTM preference. In the case of RSVP-TE tunnel type, the user can further restrict the selection by providing a list of LSP names.

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 will not be activated using IP next-hops in RTM if no tunnel next-hops are found in TTM.

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** 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** option in the system.
- 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 preferred over IP next-hops.

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 (**rsvp-shortcut** or **advertise-tunnel-link** enabled in the IGP instance).
- BGP prefix with the BGP next-hop resolved to IGP shortcuts in RTM (**rsvp-shortcut** enabled in the IGP instance).
- Static route prefix resolved to an indirect next-hop which itself 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 itself is resolved to IGP shortcuts in RTM.
- BGP prefix with a BGP next-hop resolved to a static route which itself resolves to set of tunnel next-hops towards an indirect next-hop in RTM or TTM.
- BGP prefix resolving to another BGP prefix which next-hop is resolved to set of ECMP tunnel next-hops with a static route in RTM or TTM or to IGP shortcuts in RTM.

Note that this feature does not modify the route calculation, thus the same set of ECMP next-hops is computed for a prefix. It also does not change the hash routine, but only the spraying of the flows over the tunnel next-hops is modified to reflect the normalized weight of each tunnel next-hop.

As part of this feature, static route implementation has been enhanced to support 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 on page 59.

Weighted Load Balancing IGP, BGP, and Static Route Prefix Packets over IGP Shortcut

Feature Configuration

The user must have IGP shortcut or forwarding adjacency feature enabled in one or more IGP instances:

configure>router>ospf(isis)>rsvp-shortcut

configure>router>ospf(isis)>advertise-tunnel-link

The user can also disable specific MPLS LSPs from being used in IGP shortcut or forwarding adjacency by configuring the following:

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

The user enables the weighted load balancing feature using the following new router level command:

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.

The user can configure a weight for each LSP using the following command:

configure>router>mpls>lsp>load-balancing-weight <*32-bit-integer*>

For an auto-LSP signaled via an LSP template, the weight is configured using the following command:

configure>router>mpls>lsp-template>load-balancing-weight <*32-bit-integer*>

There is no default weight value for an LSP. If one or more LSP in the ECMP set of a prefix does not have a weight configured, the regular ECMP spraying for the prefix will be performed. The user entered weight is normalized to the closest integer value which 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 will be assigned to the LSP, the more packets will be sent to this LSP.

Feature Behavior

This section describes the details of 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** 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 follows the following procedures:

1. MPLS populates the user configured LSP weight in TTM. When the global command **weighted-ecmp** is enabled, and if one or more LSPs in the ECMP set of a prefix does not have a weight configured, the regular ECMP spraying for the prefix will be 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 if 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, e.g., network failure, and updates a given route with a modified set of tunnel next-hops. This will trigger 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 will perform a route download to IOM but other users of RTM should not be notified since 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 which 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 towards a given 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 which 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 will continue to be forwarded using either regular ECMP or by selecting one next-hop from the set as in existing implementation.

ECMP Considerations

The weight assigned to an LSP impacts only the forwarding decision, not the routing decision. In other words, 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** option. This selection continues to follow the algorithm used in the IGP shortcut feature.

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

Weighted Load Balancing Static Route Packets over MPLS LSP

Feature Configuration

The user enables the resolution of a static route to a one or more MPLS P2P LSPs in TTM using the following new static route configuration command:

```
config>router>static-route-entry {ip-prefix/prefix-length} [mcast] indirect {ip-address} tunnel-next-hop
      — resolution {any|disabled|filter}
      — resolution-filter
            — [no] ldp
            — [no] rsvp-te
                  — [no] [lsp <name1>]
                  — [no] [lsp <name2>]
                  — .
                  — .
                  — [no] [lsp <namen>]
            — exit
      — [no] disallow-igp
      — exit
exit
```

The user can either provide a list of LSP names or let the automatic selection of the LSP tunnel next-hops from the TTM by configuring **resolution** to the **any** value. These are mutually exclusive. A maximum of 128 LSP names can be entered within a static route prefix configuration.

Note that a P2P auto-lsp instantiated via an LSP template can be selected in TTM when **resolution** is set to **any**. It is however 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 will black-hole traffic of the static route.

The above command is covered in much more details in Static Route Resolution Using Tunnels on page 59 which also provides the selection rules among multiple LSP types: RSVP and LDP. A given 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.

The existing **static-route** command is still supported with all other options, including the **indirect** one which can be used to resolve the indirect next-hops in RTM. The new command is an add-on to configure the resolution to tunnel next-hops in TTM. As such, the user must first configure the prefix with the existing command and the **indirect** option and then enter the new command with the indirect option and with the new **static-route-entry** command. Here is an example:

```
/configure router static-route 5.5.5.5/32 indirect 1.0.0.2
/configure router static-route-entry 5.5.5.5/32 indirect 1.0.0.2
    tunnel-next-hop
        rsvp-te
            lsp to-1.0.0.2-1
            lsp to-1.0.0.2-2
```

```
        exit
        no shutdown
    exit
```

In order 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 found in TTM to the indirect next-hop of the static route. All LSPs however must have the same LSP metric otherwise only the tunnel next-hops with the same lowest metric will be activated for the static route.

The user can force the metric of an LSP to a constant value using the following command:

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** option, only the first configured LSPs which number equals the **ecmp** value will be selected. The remaining tunnel next-hops for the route will not be activated. When automatic MPLS LSP selection is performed in TTM, the lower tunnel-id is used as a tie-breaker among the same lowest metric LSPs.

In order 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 a static, IGP and BGP prefixes resolving to IGP shortcuts:

configure>router>weighted-ecmp

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 on page 62 for the details. 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 if 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 one or more LSP in the ECMP set of a prefix static route does not have a weight configured, the regular ECMP spraying for the prefix will be 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** option in the system. In this case, the weighted load-balancing will be applied directly using the weights of the selected set of tunnel next-hops. If one or more LSP in the ECMP set of a prefix static route does not have a weight configured, or if one or more of the indirect next-hops binds to an LDP LSP, the regular ECMP spraying for the prefix will be performed.

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

Bi-directional Forwarding Detection

Bi-directional Forwarding Detection (BFD) is a light-weight, low-overhead, short-duration detection of failures in the path between two systems. If a system stops receiving BFD messages for a long enough period (based on configuration) it is assumed that a failure along the path has occurred and the associated protocol or service is notified of the failure.

BFD can provide a mechanism used for liveness detection over any media, at any protocol layer, with a wide range of detection times and overhead, to avoid a proliferation of different methods.

SR OS supports asynchronous and on deman modes of BFD in which BFD messages are set to test the path between systems.

If multiple protocols are running between the same two BFD endpoints then only a single BFD session is established, and all associated protocols will share the single BFD session.

In addition to the typical asynchronous mode, there is also an echo function defined within RFC 5880, *Bi-directional Forwarding Detection*, that allows either of the two systems to send a sequence of BFD echo packets to the other system, which loops them back within that system's forwarding plane. If a number of these echo packets are lost then the BFD session is declared down.

BFD Control Packet

The base BFD specification does not specify the encapsulation type to be used for sending BFD control packets. Instead it is left to the implementers to use the appropriate encapsulation type for the medium and network. The encapsulation for BFD over IPv4 and IPv6 networks is specified in draft-ietf-bfd-v4v6-1hop-04.txt, *BFD for IPv4 and IPv6 (Single Hop)*. This specification requires that BFD control packets be sent over UDP with a destination port number of 3784 and the source port number must be within the range 49152 to 65535.

In addition, the TTL of all transmitted BFD packets must have an IP TTL of 255. All BFD packets received must have an IP TTL of 255 if authentication is not enabled. If authentication is enabled, the IP TTL should be 255 but can still be processed if it is not (assuming the packet passes the enabled authentication mechanism).

If multiple BFD sessions exist between two nodes, the BFD discriminator is used to de-multiplex the BFD control packet to the appropriate BFD session.

Control Packet Format

The BFD control packet has 2 sections, a mandatory section and an optional authentication section.

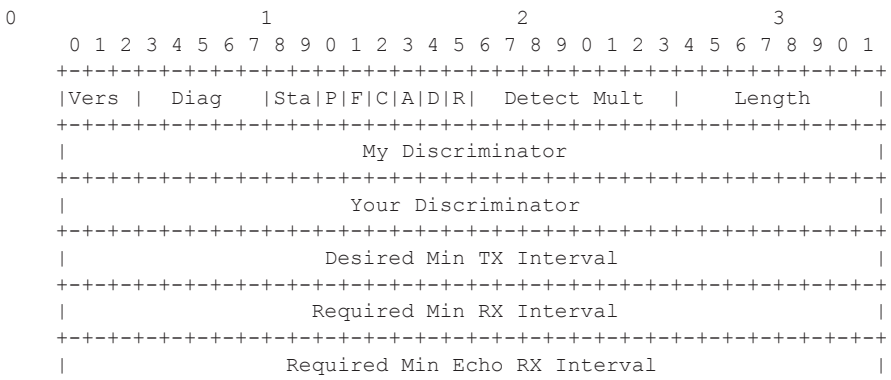

Figure 10: Mandatory Frame Format

Table 5: BFD Control Packet Field Descriptions

Field	Description
Vers	The version number of the protocol. The initial protocol version is 0.
Diag	A diagnostic code specifying the local system's reason for the last transition of the session from Up to some other state. Possible values are: 0-No diagnostic 1-Control detection time expired 2-Echo function failed 3-Neighbor signaled session down 4-Forwarding plane reset 5-Path down 6-Concatenated path down 7-Administratively down
D Bit	The "demand mode" bit. (Not supported)
P Bit	The poll bit. If set, the transmitting system is requesting verification of connectivity, or of a parameter change.
F Bit	The final bit. If set, the transmitting system is responding to a received BFD control packet that had the poll (P) bit set.
Rsvd	Reserved bits. These bits must be zero on transmit and ignored on receipt.

Table 5: BFD Control Packet Field Descriptions (Continued)

Field	Description (Continued)
Length	Length of the BFD control packet, in bytes.
My Discriminator	A unique, nonzero discriminator value generated by the transmitting system, used to demultiplex multiple BFD sessions between the same pair of systems.
Your Discriminator	The discriminator received from the corresponding remote system. This field reflects back the received value of my discriminator, or is zero if that value is unknown.
Desired Min TX Interval	This is the minimum interval, in microseconds, that the local system would like to use when transmitting BFD control packets.
Required Min RX Interval	This is the minimum interval, in microseconds, between received BFD control packets that this system is capable of supporting.
Required Min Echo RX Interval	This is the minimum interval, in microseconds, between received BFD echo packets that this system is capable of supporting. If this value is zero, the transmitting system does not support the receipt of BFD echo packets.

BFD for RSVP-TE

BFD will notify RSVP-TE if the BFD session goes down, in addition to notifying other configured BFD enabled protocols (for example, OSPF, IS-IS and PIM). This notification will then be used by RSVP-TE to begin the reconvergence process. This greatly accelerates the overall RSVP-TE response to network failures.

All encapsulation types supporting IPv4 and IPv6 is supported as all BFD packets are carried in IPv4 and IPv6 packets; this includes Frame Relay and ATM.

BFD is supported on the following interfaces:

- Ethernet (Null, Dot1Q & QinQ)
- POS interfaces (including APS)
- Channelized interfaces (PPP, HDLC, FR and ATM) on ASAP (priority 1) and channelized MDAs (Priority 2) including link bundles and IMA
- Spoke SDPs
- LAG interfaces
- VSM interfaces

Echo Support

Echo support for BFD calls for the support of the echo function within BFD. By supporting BFD echo, the router loops back received BFD echo messages to the original sender based on the destination IP address in the packet.

The echo function is useful when the local router does not have sufficient CPU power to handle a periodic polling rate at a high frequency. As a result, it relies on the echo sender to send a high rate of BFD echo messages through the receiver node, which is only processed by the receiver's forwarding path. This allows the echo sender to send BFD echo packets at any rate.

Note that the SR-OS router does not support the sending of echo requests, only the response to echo requests.

BFD Support for BGP

This feature enhancement allows BGP peers to be associated with the BFD session. If the BFD session failed, then BGP peering will also be torn down.

Centralized BFD

The following applications of centralized BFD require BFD to run on the SF/CPM.

- IES Over Spoke SDP
- BFD Over LAG and VSM Interfaces

IES Over Spoke SDP

One application for a central BFD implementation is so BFD can be supported over spoke SDPs used to inter-connection IES or VPRN interfaces. When there are spoke SDPs for inter-connections over an MPLS network between two routers, BFD is used to speed up failure detections between nodes so re-convergence of unicast and multicast routing information can begin as quickly as possible.

The MPLS LSP associated with the spoke SDP can enter or egress from multiple interfaces on the box. BFD for these types of interfaces can not exist on the IOM itself.

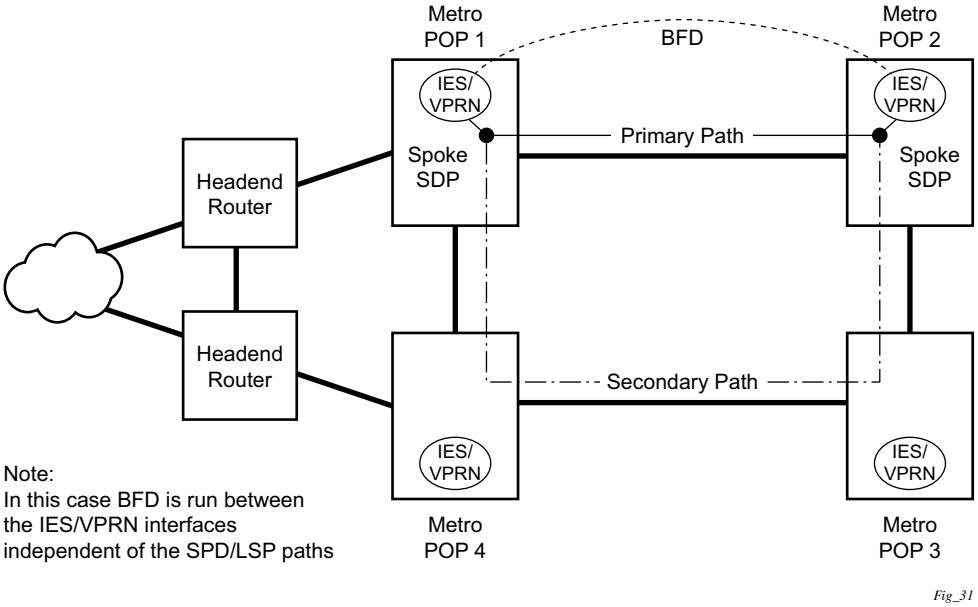

Figure 11: BFD for IES/VPRN over Spoke SDP

BFD Over LAG and VSM Interfaces

A second application for a central BFD implementation is so BFD can be supported over LAG or VSM interface. This is useful where BFD is not used for link failure detection but instead for node failure detection. In this application, the BFD session can run between the IP interfaces associated with the LAG or VSM interface, but there is only one session between the two nodes. There is no requirement for the message flow to across a certain link, or VSM, to get to the remote node.

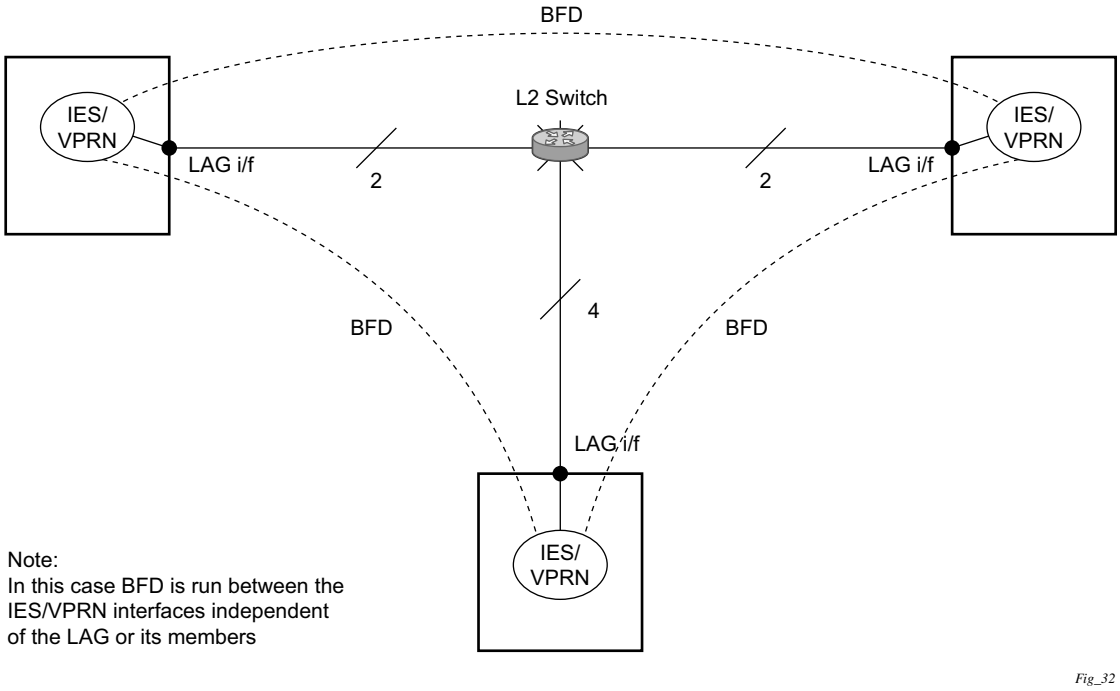

Figure 12: BFD over LAG

Aggregate Next Hop

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

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 due to 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.

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 a specific flag is added to the static route.

static-route {*ip-prefix/prefix-length*| *ip-prefix netmask* } **next-hop** *ip-int-name*|*ip-address* **validate-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 normal ARP who-has procedure, the static route becomes valid/active and is installed.

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 a specific flag is added to the static route.

configure router static-route 2001:db8::/64 next-hop 2001:db8:abba::2 validate-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 normal NS/NA procedure, the static route becomes valid/active and is installed.

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.

config>router>ldp-shortcut [ipv4][ipv6]

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 a given 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 given 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 option is enabled globally in LDP *ldp-interarea-prd*.

Note that the LDP next-hop entry is not exported to LDP control plane or to any other control plane protocols except OSPF, IS-IS, and specific OAM control plane as specified in Handling of Control Packets on page 78.

This feature is not restricted to /32 IPv4 prefixes or /128 IPv6 FEC prefixes. However only /32 IPv4 and /128 IPv6 FEC prefixes will be 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 will be forwarded over the LDP LSP. The following is an example of the resolution process.

Assume 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. Once LDP activated the FEC, it programs the NHLFE in the egress data path and the LDP tunnel information in the ingress data path tunnel table.

Next, LDP provides the shortcut route to RTM which will associate it with the same /24 prefix. There will be 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 which succeed a longest prefix match against the /24 route entry to use the LDP LSP.

Assume now 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 but it will still not have a specific /24 IP route entry. RTM then resolves all user prefixes which succeed a longest prefix match against the /24 route entry to use the LDP LSP while all other prefixes which succeed a longest prefix-match against the /16 route entry will use the IP next-hop. LDP shortcut will also work when using RIP for routing.

LDP Shortcut Forwarding Plane

Once LDP activated a FEC for a given prefix and programmed RTM, it also programs the ingress Tunnel Table in IOM 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 will result 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 unlabelled.

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 due to LDP control plane issues) the switchover should be faster. If the next-hop determination requires IGP to re-converge, this will take longer. However no target is set.

The switching from a regular IP next-hop to an LDP shortcut next-hop will normally occur 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 tunnel table are asynchronous. If Tunnel Table is configured first, it is possible that traffic will be black holed for some time .

ECMP Considerations

When ECMP is enabled and multiple equal-cost next-hops exit for the IGP route, the ingress IOM will spray the packets for this route based on hashing routine currently supported for IPv4 packets.

When the preferred RTM entry corresponds to an LDP shortcut route, spraying will be 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 in existing implementation.

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

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

Handling of Control Packets

All control plane packets will not see the LDP shortcut route entry in RTM with the exception of the following control packets which will be 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 will continue to be forwarded over the IP next-hop route in RTM.

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 interfaces in PIM. Only an RSVP P2MP LSP is currently allowed.

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

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 will continue to resolve a BGP next-hop to an LDP shortcut if the user enabled the following option in BGP:

```
config>router>bgp>next-hop-resolution>shortcut-tunnel
          family ipv4
               resolution-filter ldp
```

Interaction with Static Route Resolution to an LDP FEC

A static route will continue to be resolved by sarching an LDP LSP which 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 will be selected and if both a static and IGP routes exist, the RTM route type preference will be used to select one.

LDP Control Plane

In order for the LDP shortcut to be usable, an SR-OS router 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. In other words, it must assume it is an egress LER for the FEC until the route disappears from the routing table or the next-hop advertised a binding for the FEC prefix. In the latter case, the SR-OS router becomes a transit LSR for the FEC.

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

You must 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.

Process Overview

The following items are components to configure basic router parameters.

- Interface — A logical IP routing interface. Once 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) Creates confederation autonomous systems within an AS to reduce the number of IBGP sessions required within an AS.

Configuration Notes

The following information describes router configuration caveats.

- A system interface and associated IP address should be specified.
- Boot options file (BOF) parameters must be configured prior to configuring router parameters.
- Confederations can be configured before protocol connections (such as BGP) and peering parameters are configured.
- IPv6 interfaces and associated routing protocols may only be configured on the following systems:
  - → Chassis systems running in chassis mode c or d.
  - → Chassis systems running in mixed-mode with IPv6 functionality limited to those interface on slots with IOM3-XPs/IMMs or later line cards.
  - → 7750 SR-c4/12.
- An iom2-20g and a SFM2 card are required to enable the IPv6 CPM filter and per-peer queuing functionality.