Chapter 2
Configuring MPLS

This chapter contains the following sections:

Overview

Multiprotocol Label Switching (MPLS) is a hybrid protocol that integrates network layer routing with label switching to provide a layer 3 network with traffic management capability. MPLS provides traffic-engineering capabilities that make effective use of network resources while maintaining high bandwidth and stability. MPLS enables service providers to provide their customers with the best service available given the provider's resources, with or without traffic engineering. MPLS is the foundation for layer 3 and layer 2 VPNs.

The two basic components of MPLS are label distribution and data mapping.

- Label distribution is the set of actions MPLS performs to establish and maintain a label-switched path (LSP), also known as an *MPLS tunnel*.
- Data mapping is the process of getting data packets onto an established LSP.

Conventions in This Chapter

Certain terms used with MPLS, such as the names of messages, are often expressed in the RFCs and other sources either with initial uppercase letters or all uppercase letters. For improved readability, those terms are represented in lowercase in this chapter. Table 23 lists the terms and some of their variant spellings.

Table 23: Conventions for MPLS Terms

In This Chapter	In RFCs and Other Sources	
ack	Ack	ACK
bundle	Bundle	–
hello	Hello	HELLO
initialization	Initialization	INITIALIZATION
keepalive	Keepalive	KEEPALIVE
label mapping	Label Mapping	LABEL_MAPPING
label release	Label Release	LABEL_RELEASE
label request	Label Request	LABEL_REQUEST
label request abort	Label Request Abort	LABEL_REQUEST_ABORT
label withdrawal	Label Withdrawal	LABEL_WITHDRAWAL
message ack	message_Ack	MESSAGE_ACK
message ID	message_ID	MESSAGE_ID
srfresh	Srefresh	–

Table 23: Conventions for MPLS Terms (continued)

In This Chapter	In RFCs and Other Sources	
path	Path	PATH
patherr	PathErr	PATHERR
pathtear	PathTear	PATHTEAR
resv	Resv	RESV
resvconf	ResvConf	RESVCONF
resverr	ResvErr	RESVERR
resvtear	ResvTear	RESVTEAR
targeted hello	Targeted Hello	TARGETED_HELLO

MPLS Terms and Acronyms

Table 24 defines terms and acronyms that are used in this discussion of MPLS.

Table 24: MPLS Terms and Acronyms

Term	Definition
Admission control	Accounting mechanism that tracks resource information. Prevents requests from being accepted if sufficient resources are not available.
BGP	Border Gateway Protocol, which provides loop-free interdomain routing between autonomous systems (ASs) and can act as a label distribution protocol for MPLS.
Constraint-based routing	A mechanism to establish paths based on certain criteria (explicit route, QoS parameters). The standard routing protocols can be enhanced to carry additional information to be used when running the route calculation.
E-LSP	EXP-inferred-PSC LSP. The EXP field of the MPLS Shim Header is used to determine the per-hop behavior applied to the packet.
Explicit routing	A subset of constraint-based routing where the constraint is an explicit route
FEC	Forwarding equivalence class—Group of IP packets forwarded over the same path with the same path attributes applied
Label Distribution Protocol	■ A particular label distribution protocol used for label distribution among the routers in an MPLS domain; represented by the acronym LDP ■ In lowercase—label distribution protocol—a generic term for any of several protocols that distribute labels among the routers in an MPLS domain, including BGP, LDP, and RSVP-TE. This usage is **not** represented in this text by the acronym, LDP.
LDP	Label Distribution Protocol—A particular protocol used for label distribution among the routers in an MPLS domain This text does **not** use LDP to refer to the generic class of label distribution protocols.
LER	Label edge router—A label-switching router serving as an ingress or egress nodes
LSP	Label-switched path—The path traversed by a packet that is routed by MPLS. Some LSPs act as tunnels.

Table 24: MPLS Terms and Acronyms (continued)

Term	Definition
LSP priority level	A priority that indicates the importance of one LSP relative to another LSP. LSPs having higher priorities can preempt LSPs having lower priorities. Priorities range from 0 through 7 in order of *decreasing* priority.
L-LSP	Label-only-inferred-PSC LSP. The label value, and possibly the EXP-bits, are used to determine the per-hop behavior applied to the packet.
LSR	Label-switching router—An MPLS node that can forward layer 3 packets based on their labels
MPLS	Multiprotocol Label Switching—Set of techniques enabling forwarding of traffic using layer 2 and layer 3 information
MPLS edge node	MPLS node that connects an MPLS domain with a node outside the domain that either does not run MPLS or is in a different domain
MPLS egress node	MPLS edge node in the role of handling traffic as it leaves an MPLS domain
MPLS ingress node	MPLS edge node in the role of handling traffic as it enters an MPLS domain
MPLS label	Label carried in a packet header that represents a packet's forwarding equivalence class
MPLS node	A router running MPLS. An MPLS node is aware of MPLS control protocols, operates one or more L3 routing protocols, and is capable of forwarding packets based on labels. Optionally, an MPLS node can be capable of forwarding native L3 packets.
Provider edge router	PE—An LER at the edge of a service provider core that provides ingress to or egress from a VPN
Provider core router	P—An LSR within a service provider core that carries traffic for a VPN
RSVP	Resource Reservation Protocol; E-series routers do not support RSVP
RSVP-TE	Resource Reservation Protocol enhanced to support MPLS traffic engineering; E-series routers support RSVP-TE
Traffic engineering	The ability to control the path taken through a network or portion of a network based on a set of traffic parameters (bandwidth, QoS parameters, and so on). Traffic engineering (TE) enables performance optimization of operational networks and their resources.
Tunnel	LSP that is used by an IGP to reach a destination, or an LSP that uses traffic engineering

Features

The following major features are currently supported by MPLS:

- BFD fast failure detection for RSVP-TE adjacencies
- Differentiated services
- Interface support
  - ATM AAL5 (RSVP-TE only)
  - ATM1483 (point-to-point AAL5SNAP only)
  - Ethernet/VLAN
  - GRE
  - Multilink PPP
  - POS (PPP over HDLC)
  - PPP
  - SLEP (Cisco HDLC)
- Label stacking
  - Virtual Private Networks (VR-based and BGP-based)
  - Layer 2 Services over MPLS
- LER functionality
- LSR functionality
- Spoof checking
- LDP graceful restart
- ECMP
- Topology-driven LSPs (LDP) including support of LDP over RSVP tunnels
- Traffic engineering (RSVP-TE)
  - Constraint-based explicit routing
  - Statically configured explicit routing
  - Hop-by-hop routing
  - Admission control and bandwidth enforcement
  - Tunnels used by IGP as next hops in SPF calculation

  - Tunnel ingress controlled failure recovery
  - Facility back-up-style fast-route
- Traffic support
  - Layer 2 frames: ATM, Ethernet, Frame Relay, HDLC, PPP, VLAN
  - Layer 3 datagrams: IPv4, IPv6

Platform Considerations

For information about modules that support MPLS on the ERX-7xx models, ERX-14xx models, and the ERX-310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support BGP.

For information about modules that support MPLS on E120 routers and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support MPLS.

References

For more information about the MPLS protocol, consult the following resources:

- *JUNOSe Release Notes*, *Appendix A*, *System Maximums*—Refer to the Release Notes corresponding to your software release for information about maximum values.
- BGP/MPLS IP VPNs—draft-ietf-l3vpn-rfc2547bis-03.txt (April 2005 expiration)
- Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)—draft-ietf-mpls-in-ip-or-gre-03.txt (September 2003 expiration)
- LDP IGP Synchronization—draft-jork-ldp-igp-sync-01.txt (August 2005 expiration)
- RFC 2104—HMAC: Keyed-Hashing for Message Authentication (February 1997)
- RFC 2205—Resource ReSerVation Protocol (RSVP) -- Version 1, Functional Specification (September 1997)
- RFC 2209—Resource ReSerVation Protocol (RSVP) -- Version 1, Message Processing Rules (September 1997)

- RFC 2210—The Use of RSVP with IETF Integrated Services (September 1997)
- RFC 2211—Specification of the Controlled-Load Network Element Service (September 1997)
- RFC 2474—Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (December 1998)
- RFC 2475—An Architecture for Differentiated Services (December 1998)
- RFC 2547—BGP/MPLS VPNs (March 1999)
- RFC 2597—Assured Forwarding PHB Group (June 1999)
- RFC 2685—Virtual Private Networks Identifier (September 1999)
- RFC 2702—Requirements for Traffic Engineering over MPLS (September 1999)
- RFC 2747—RSVP Cryptographic Authentication (January 2000)
- RFC 2836—Per Hop Behavior Identification Codes (May 2000)
- RFC 2858—Multiprotocol Extensions for BGP-4 (June 2000)
- RFC 2961—RSVP Refresh Overhead Reduction Extensions (April 2001)
- RFC 3031—Multiprotocol Label Switching Architecture (January 2001)
- RFC 3032—MPLS Label Stack Encoding (January 2001)
- RFC 3035—MPLS using LDP and ATM VC Switching (January 2001)
- RFC 3036—LDP Specification (January 2001)
- RFC 3037—LDP Applicability (January 2001)
- RFC 3097—RSVP Cryptographic Authentication -- Updated Message Type Value (April 2001)
- RFC 3107—Carrying Label Information in BGP-4 (May 2001)
- RFC 3140—Per Hop Behavior Identification Codes (June 2001)
- RFC 3209—RSVP-TE: Extensions to RSVP for LSP Tunnels (December 2001)
- RFC 3210—Applicability Statement for Extensions to RSVP for LSP-Tunnels (December 2001)
- RFC 3246—An Expedited Forwarding PHB (Per-Hop Behavior) (March 2002)
- RFC 3270—Multi-Protocol Label Switching (MPLS) Support of Differentiated Services (May 2002)
- RFC 3443—Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks (January 2003)

- RFC 3471—Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description (January 2003)
- RFC 3473—Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions (January 2003)
- RFC 3478—Graceful Restart Mechanism for Label Distribution Protocol (February 2003)
- RFC 3479—Fault Tolerance for the Label Distribution Protocol (LDP) (February 2003)
- RFC 3564—Requirements for support of Differentiated Services-aware MPLS Traffic Engineering (July 2003)
- RFC 4090—Fast Reroute Extensions to RSVP-TE for LSP Tunnels (May 2005)
- RFC 4379—Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures (February 2006)

NOTE: IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Web site at http://www.ietf.org for the latest drafts.

How MPLS Works

This section describes elements of MPLS and how they interact.

IP Routing

In conventional IP routing, as a packet traverses from one router to the next through a network, each router analyzes the packet's header and performs a network layer routing table lookup to choose the next hop for the packet. In conventional IP forwarding, the router looks for the address in its forwarding table with the longest match (best match) for the packet's destination address. All packets forwarded to this longest match are considered to be in the same forwarding equivalence class (FEC).

Label Switching

MPLS is not a routing protocol; it works with layer 3 routing protocols (BGP, IS-IS, OSPF) to integrate network layer routing with label switching. An MPLS FEC consists of a set of packets that are all forwarded in the same manner by a given label-switching router (LSR). For example, all packets received on a particular interface might be assigned to a FEC. MPLS assigns each packet to a FEC only at the LSR that serves as the ingress node to the MPLS domain. A label distribution protocol binds a label to the FEC. Each LSR uses the label distribution protocol to signal its forwarding peers and distribute its labels to establish an LSP. The label distribution protocol enables negotiation with the downstream LSRs to determine what labels are used on the LSP and how they are employed.

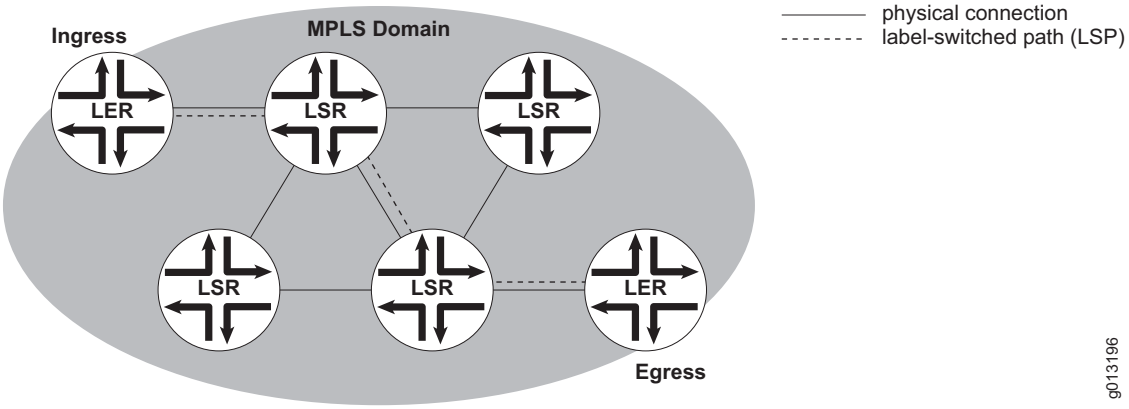
Labels represent the FEC along the LSP from the ingress node to the egress node. The label is prepended to the packet when the packet is forwarded to the next hop. Each label is valid only between a pair of LSRs. A downstream LSR reached by a packet uses the label as an index into a table that contains both the next hop and a different label to prepend to the packet before forwarding. This table is usually referred to as a label information base (LIB).

The LSR that serves as the egress MPLS node uses the label as an index into a table that has the information necessary to forward the packet from the MPLS domain. The forwarding actions at the egress LSR can be any of the following:

- Forward the packet based on the inner header exposed after popping the label. This can be accomplished either by doing a routing table lookup or forwarding based on the exposed inner MPLS label.
- Forward the packet to a particular neighbor as directed by the table entry, for example in a Martini layer 2 transport case.

Figure 47 shows a simple MPLS domain, consisting of multiple LSRs. The LSRs serving as ingress and egress nodes are also referred to as *label edge routers* (LERs). The ingress router is sometimes referred to as the *tunnel head end*, or the *head-end* router. The egress router is sometimes referred to as the *tunnel tail end*, or the *tail-end* router. LSPs are *unidirectional*, carrying traffic only in the downstream direction from the ingress node to the egress node.

Figure 47: Simple MPLS Domain



Label-Switching Routers

Each LSR, also known as an MPLS node, must have the following:

- At least one layer 3 routing protocol
- A label distribution protocol
- The ability to forward packets based on their labels

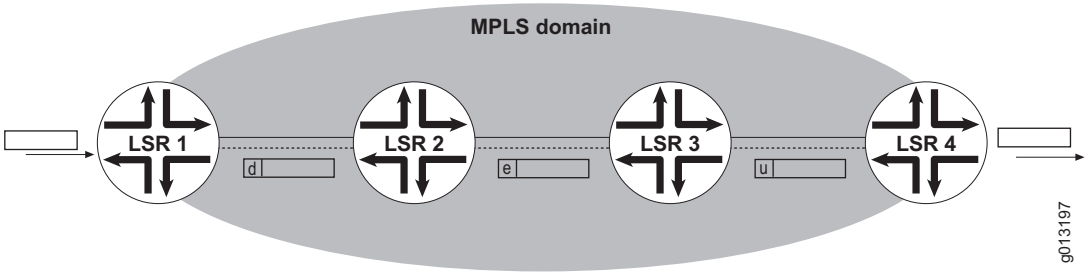
The router can use BGP, IS-IS, or OSPF as its layer 3 routing protocol, and BGP, LDP, or RSVP-TE as its label distribution protocol.

Label Switching and Popping

MPLS can label packets by using the existing layer 2 header or an encapsulation header that carries the MPLS label. During LSP negotiation, the LSRs in an MPLS domain agree on a labeling method. Labels have only local meaning; that is, meaning for two LSR peers. Each pair of LSRs—consisting of a label originator and a label acceptor—must use a label distribution protocol to agree on the label-to-FEC binding.

Because of the local label assignment, packet labels typically change at each segment in the LSP path, as shown in Figure 48. The ingress node, LSR 1, receives an unlabeled data packet and prepends label *d* to the packet. LSR 2 receives the packet, removes label *d* and uses it as an index in its forwarding table to find the next label. LSR 2 prepends label *e* to the packet. LSR 3 does the same thing, removing label *e* and prepending label *u*. Finally, the egress node, LSR 4, removes label *u* and determines where to forward the packet outside the MPLS domain.

Figure 48: Label Switching



Any packet can carry multiple labels. The labels are stacked in a last-in-first-out order. Each LSR forwards packets based on the outermost (top) label in the stack. An LSR *pushes* a label onto the stack when it prepends the label to a packet header. It *pops* the label when it pulls the label off the stack and compares it with the forwarding table. On determining the label for the next segment of the LSP, the LSR pushes the new label on the stack. A label swap consists of a pop, lookup, and push.

When the egress router, such as LSR 4 in Figure 48, receives a packet, it may perform two lookups: it looks up the label and determines that the label must be popped, then it does another lookup based on the exposed header to determine where to forward the packet. This behavior is known as *ultimate hop popping*, and was the only possible action for the JUNOSe implementation before Release 7.3.0.

Beginning with JUNOSe Release 7.3.0, an alternative behavior, known as *penultimate hop popping* (PHP), is the default when RSVP-TE is the signaling protocol. Beginning with JUNOSe Release 8.1.0, PHP is also the default when LDP is the signaling protocol. PHP reduces the number of lookups performed by the LER. In PHP, the LER requests its upstream neighbor (the penultimate hop) to pop the outermost label and send just the packet to the LER. The LER then performs only the lookup for the packet. The request to perform PHP is signaled by the LER when it includes an implicit null label in the label mapping message that it sends to its upstream neighbor. The implicit null label never appears in the encapsulation.

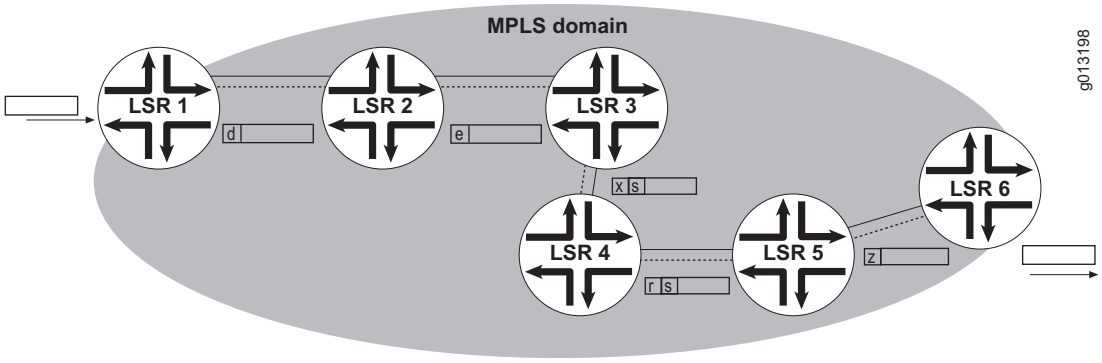
You can still achieve ultimate hop popping by configuring the egress router to advertise an explicit null label to its upstream neighbor. This advertisement, performed by LDP or RSVP-TE, ensures that all MPLS packets traversing the LSP to the egress router include a label. Alternatively, you can configure the egress router to advertise real (non-null) labels, and achieve the same result.

Regardless of whether the LSR advertises the implicit null label to achieve PHP on an upstream neighbor, if the LSR receives a PHP request from a downstream neighbor, then the LSR does perform the PHP for its neighbor.

Label Stacking

Figure 49 shows an LSP that uses label stacking. The ingress node, LSR 1, receives an unlabeled data packet and prepends label *d* to the packet. LSR 2 receives the packet, removes label *d* and uses it as an index in its forwarding table to find the next label, prepending label *e* to the packet. LSR 3 removes label *e* and prepends label *s* (negotiated with LSR 5) to the packet. LSR 3 pushes label *x* on top of label *s*. LSR 4 pops the top (outermost) label, *x*, and pushes label *r* on top of label *s*. LSR 5 pops label *r*, determines that it must pop label *s*, and pushes label *z* on the empty stack. Finally, the egress node, LSR 6, removes label *z* and determines where to forward the packet outside the MPLS domain.

Figure 49: Label Stacking



The configuration shown in Figure 49 is an example of an LSP within an LSP (a tunnel within a tunnel). The first LSP consists of LSR 1, LSR 2, LSR 3, LSR 5, and LSR 6. The second LSP consists of LSR 3, LSR 4, and LSR 5. The two LSPs have different ingress and egress points. LSR 1 and LSR 6 are LERs. Less obviously, LSR 3 and LSR 5 are also LERs, but for the internal LSP.

NOTE: Label stacking is typically employed for LSR peers that are not directly connected. Figure 49 is a simplified example to illustrate the concept of label stacking.

Labels

MPLS uses labels from either the *platform label space* or the *interface label space*. ATM AAL5 interfaces always use labels from only the interface label space. For every interface using the interface label space, you must define the range available to the router for labels in the interface label space. All other interface types always use labels from only the platform label space. You cannot configure the range for the platform label space.

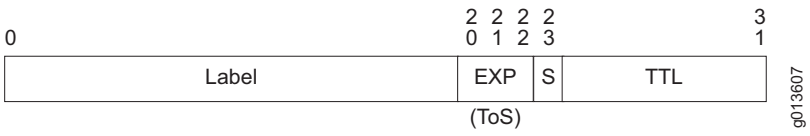
The platform label space is a large, single, unconfigurable pool of labels that can be shared by the platform—all MPLS interfaces on a given virtual router. By contrast, interface labels enable you to effectively create multiple smaller pools of labels, each used only by a particular interface. When you configure interface labels, you restrict only a given interface to a particular range of labels. Other interfaces in that VR can still use labels from that space unless you restrict them in turn to a different range of interface labels.

In the interface label space, MPLS selects labels from interface resources, a VPI/VCI combination. You configure a VPI range and a VCI range available to the labels. When an upstream LSR requests a label, the downstream LSR allocates a VPI/VCI combination for use as a label between these two peers. Allocating labels on a per interface basis is necessary because the VPI/VCI ranges are limited. This enables you to use the same label on different interfaces without conflict.

When you use the platform label space, the MPLS ingress node places labels in *shim headers* between the link-layer header and the payload. The shim header includes the following bits (Figure 50):

- Label bits—Twenty bits
- EXP bits—Three bits for class of service information; these bits are variously called the experimental bits, class of service (CoS) bits, or type of service (ToS) bits. The EXP bits are mapped from the IP packet at the ingress node and are mapped back into the IP packet at the egress node.
- S bit—One bit to indicate whether the label is on the bottom of the label stack.
- TTL bits—Eight bits for a time-to-live indicator. The TTL bits are mapped from the IP packet at the ingress node. The TTL bits in the shim header are decremented at each hop. The bits are mapped back into the IP packet at the egress node. See *TTL Processing in the Platform Label Space* on page 203 for more information.

Figure 50: Shim Header



If you configure an MPLS interface to use the interface label space, the VPI/VCI combinations are used as labels, so there is no need to place them within a shim header. As the data travels along the LSP, the LSRs examine only the VPI/VCI combination. The shim header is used only to carry the TTL bits to the egress, and is not visible to intermediate LSRs. The ingress node learns the total hop count from signaling and then uses that count to decrement the TTL to the correct final value. The TTL is then carried in the shim header to the egress node without modification, arriving with the correct count.

TTL Processing in the Platform Label Space

JUNOSe MPLS TTL processing is compliant with RFC 3443. The details of TTL processing vary with the tunnel model that is configured for TTL processing, pipe or uniform.

To keep backward compatibility with earlier JUNOSe releases, you do not use the **mpls tunnel-model** command to configure the tunnel model for TTL processing, That command is used instead to configure the tunnel model for EXP bits processing. The default tunnel model varies between TTL and EXP processing; for EXP processing, the default tunnel model is pipe, while for TTL processing the default tunnel model is uniform.

You can issue the **no mpls ip propagate-ttl** command to change the TTL processing tunnel model from the default uniform model to the pipe model. Issue the **no mpls ip propagate-ttl local** command to set the tunnel model to pipe for locally originated packets. Issue the **no mpls ip propagate-ttl forwarded** command to set the tunnel model to pipe for forwarded packets.

TTL Processing on Incoming MPLS Packets

The flow chart on page 204 (Figure 51) illustrates TTL processing on incoming MPLS packets. On a transit LSR or an egress LER, MPLS pops one or more labels and can push one or more labels. The incoming TTL of the packet is determined by the configured TTL processing tunnel model.

When all of the following conditions are met, the incoming TTL is set to the TTL value found in the immediate inner header:

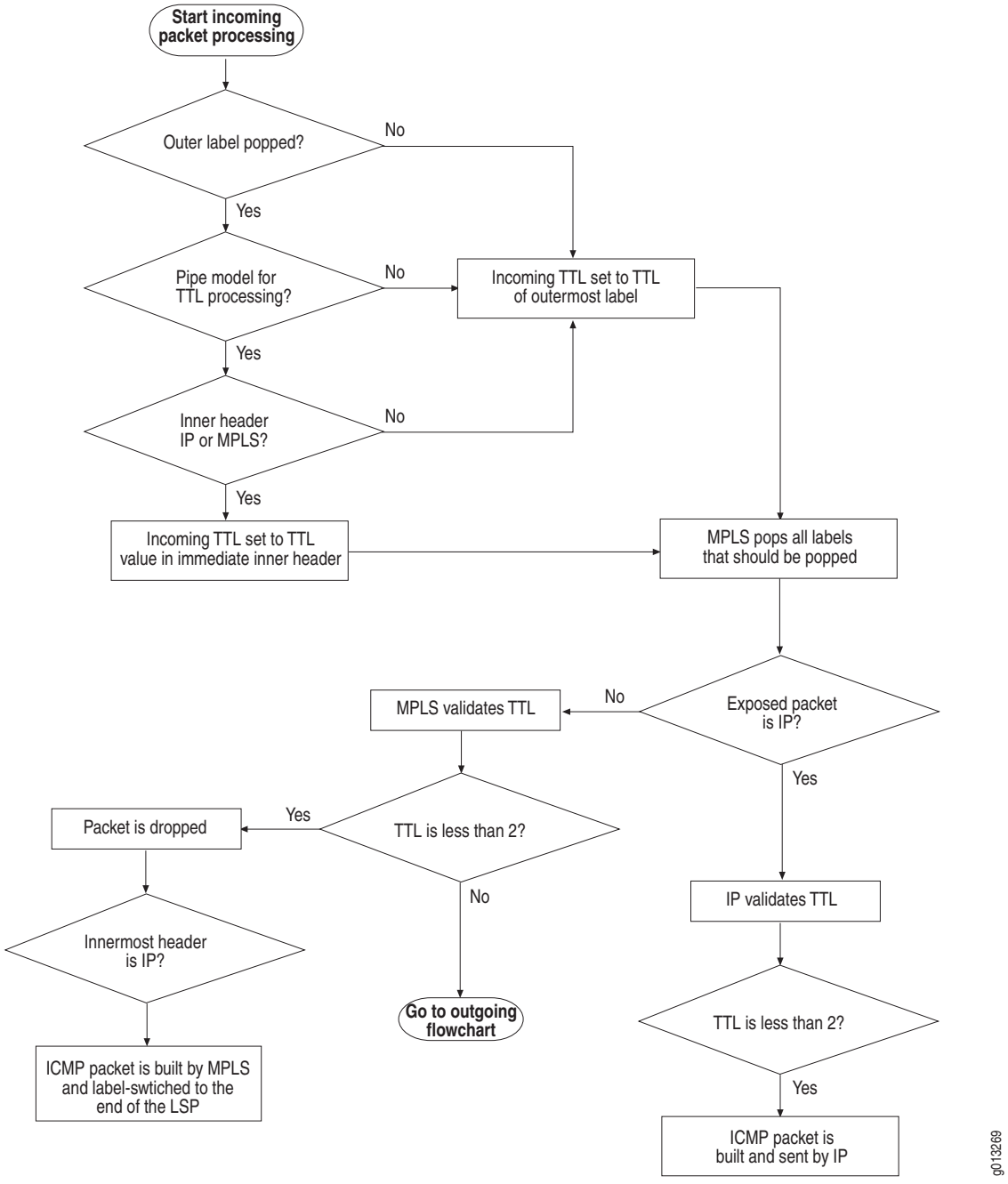
- The outer label is popped as opposed to being swapped
- The TTL processing model is configured to pipe
- The inner header is MPLS or IP

If any of those conditions is not met, then the incoming TTL is set to the TTL value found in the outermost label. In all cases, the TTL values of any further inner labels are ignored.

When an IP packet is exposed after MPLS pops all the labels that should be popped, MPLS passes the packet to IP for further processing, including TTL checking. When the uniform tunnel model for TTL processing is in effect, MPLS sets the TTL value of the IP packet to the incoming TTL value that was just set. In other words, the TTL value is copied from the outermost label to the IP packet. When the pipe model for TTL processing is in effect, the TTL value in the IP header is left unchanged.

If an IP packet is not exposed by the label popping, then MPLS performs the TTL validation. If the incoming TTL is less than 2, the packet is dropped. If innermost packet is IP, an ICMP packet is built and sent. If the TTL does not expire and the packet needs to be sent out, the outgoing TTL is determined by the rules for outgoing MPLS packets.

Figure 51: TTL Processing on Incoming MPLS Packets



TTL Processing on Outgoing MPLS Packets

The flow chart on page 206 (Figure 52) illustrates TTL processing on outgoing MPLS packets.

Rules for Processing on an LSR

On an LSR—where an MPLS packet is label-switched after processing on the line module—the TTL value in the swapped-to label is decremented by 1 from the incoming TTL value when the swapped-to label is not implicit-null. When the swapped-to label is implicit-null (for example, in a PHP configuration), the inner or exposed header's TTL is either left unchanged (when the **forwarded** option for the **mpls ip propagate-ttl** command has been configured) or is decremented by 1 from the incoming TTL value. If MPLS needs to push more labels, it sets the TTL for each label according to the following LER rules, because for those labels the router effectively is an ingress LER.

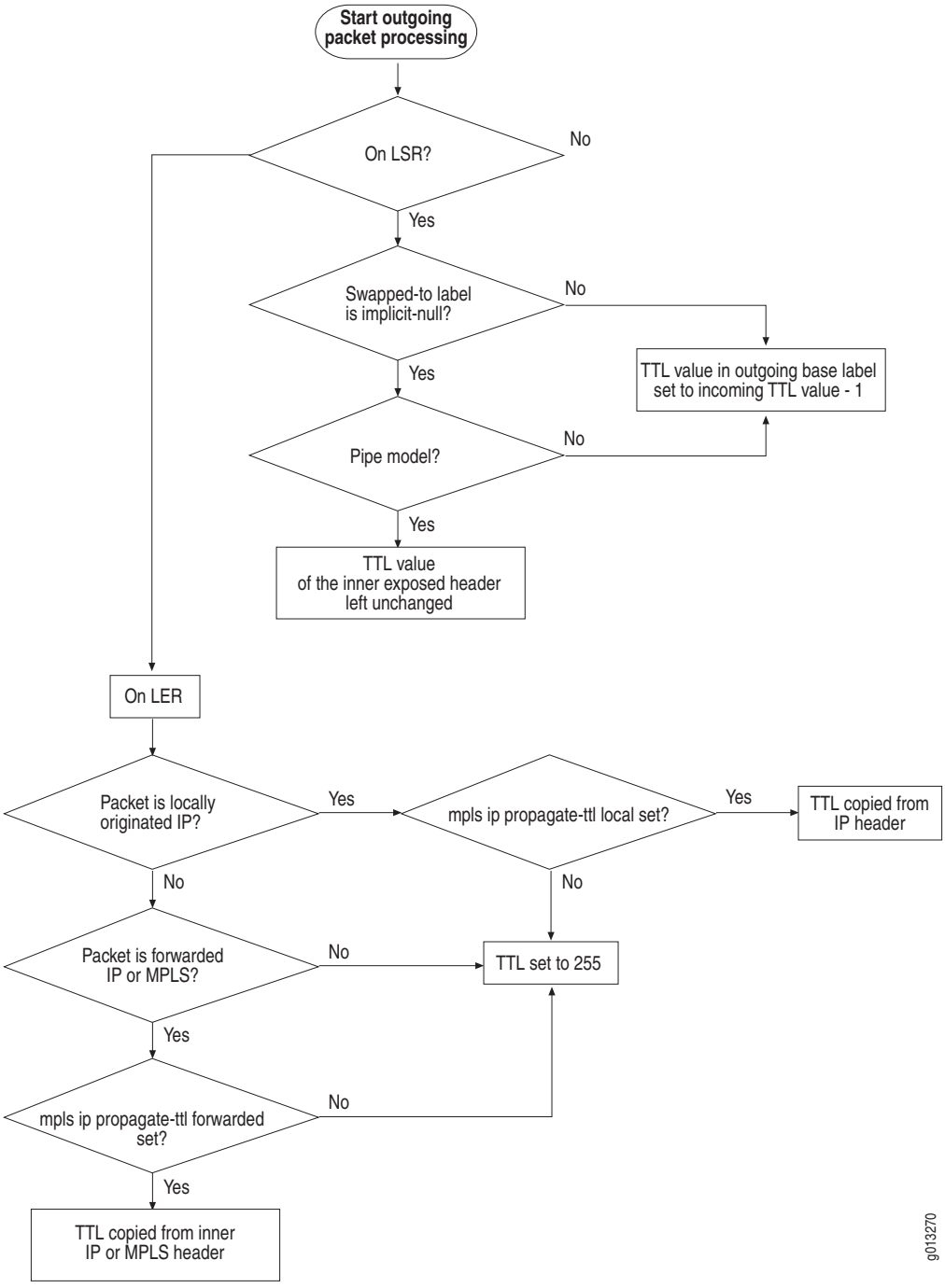
Rules for Processing on an LER

On an LER, when the packet is a locally originated IP packet, MPLS copies the TTL of all pushed MPLS labels from the IP header when the **local** option for the **mpls ip propagate-ttl** command has been configured. When the **no mpls ip propagate-ttl local** command has been configured, MPLS sets the TTL to 255.

When the packet is a forwarded IP or MPLS packet, MPLS copies the TTL of all pushed labels from the inner IP or MPLS header when the **forwarded** option for the **mpls ip propagate-ttl** command has been configured. When the **no mpls ip propagate-ttl forwarded** command has been configured, MPLS sets the TTL for these pushed labels to 255.

When the packet is neither IP nor MPLS, such as a Martini packet, MPLS sets the TTL of all pushed labels to 255.

Figure 52: TTL Processing on Outgoing MPLS Packets



MPLS Rules for TTL Expiration

MPLS takes the following actions when the TTL in a MPLS label of a received MPLS packet expires:

1. A TTL-expired ICMP packet is constructed.

2. The destination address of ICMP packet is set to the source address of the IP packet that was encapsulated in the MPLS packet.

3. The source address of ICMP packet is set to the router ID of the router on which the TTL expired.

4. The first 128 bytes of the MPLS packet including the IP payload encapsulated in the MPLS packet are copied into the payload of the ICMP packet, followed by the entire label stack of the original packet.

The ICMP packet is label-switched to the end of the LSP. From that location, the packet is forwarded back to the source of the IP packet. This behavior enables IP trace-route to work even when the LSR in the middle of the LSP does not have an IP route to the source address of the IP packet.

Label Distribution Methodology

The JUNOSe implementation of MPLS supports the following methods of label distribution:

- Downstream-on-demand, ordered control with RSVP-TE

- Downstream-unsolicited, independent control or ordered control with LDP; ordered control is the default. BGP accepts only downstream-unsolicited, ordered control

Downstream-on-demand means that MPLS devices do not signal a FEC-to-label binding until requested to do so by an upstream device. Upstream is the direction toward a packet's source; the ingress node in an MPLS domain is the farthest possible upstream node. Downstream is the direction toward a packet's destination; the egress node in an MPLS domain is the farthest possible downstream node. The egress node is sometime referred to as the *tunnel endpoint*.

Downstream-on-demand conserves labels in that they are not bound until they are needed and the LSR receives label mappings (also known as label bindings) from a neighbor that is the next hop to a destination; it is used when RSVP is the signaling protocol.

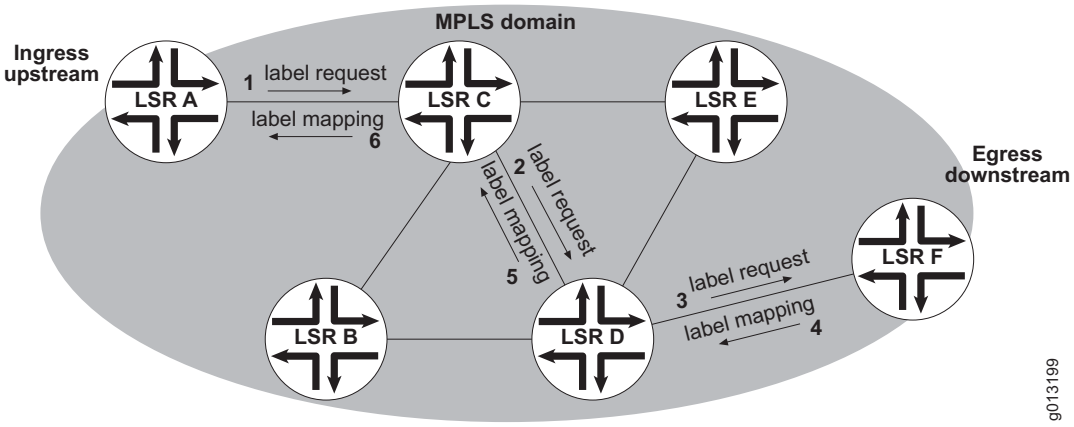
Ordered control means that an LSR does not advertise a label for a FEC unless it is the egress LSR for the FEC or until it has received a label for the FEC from its downstream peer. In this manner the entire LSP is established before MPLS begins to map data onto the LSP, preventing inappropriate (early) data mapping from occurring on the first LSR in the path.

An LSR is an egress LSR for a FEC when the FEC is its directly attached interface or when MPLS is not configured on the next-hop interface.

In Figure 53, LSR A sends a label request to LSR C. Before LSR C responds, it sends its own request to LSR D. LSR D in turn makes a request for a label to LSR F. When LSR F returns an acceptable label to LSR D, that label is for use only between LSRs D and F. LSR D sends a label back to LSR C that this pair of LSRs will use. Finally, LSR C sends back to LSR A the label that they will use. This completes the establishment of the LSP.

Downstream-unsolicited means that MPLS devices do not wait for a request from an upstream device before signaling FEC-to-label bindings. As soon as the LSR learns a route, it sends a binding for that route to all peer LSRs, both upstream and downstream. Downstream-unsolicited does not conserve labels, because an LSR receives label mappings from neighbors that might not be the next hop for the destination; it is used by BGP or LDP when adjacent peers are configured to use the platform label space.

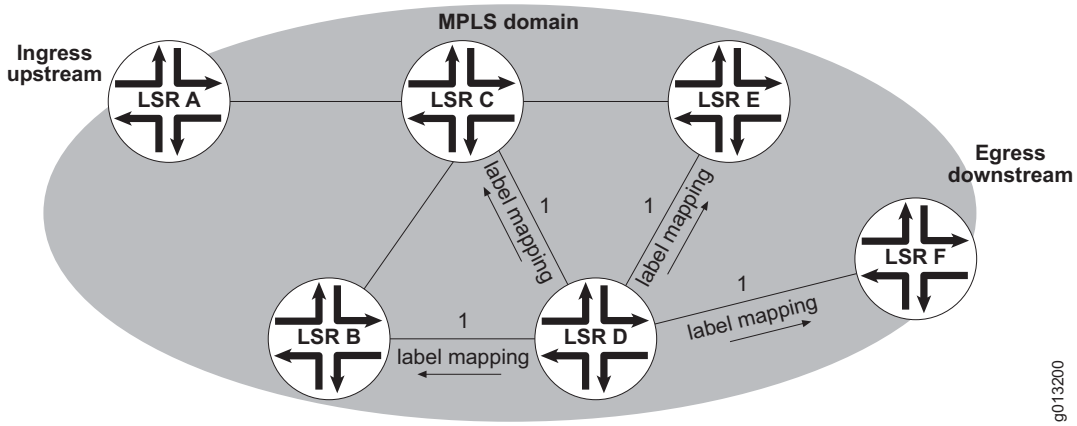
Figure 53: LSP Creation, Downstream-on-Demand, Ordered Control



Independent control means that the LSR sending the label acts independently of its downstream peer. It does not wait for a label from the downstream LSR before it sends a label to its peers. When an LSR advertises a label to an upstream neighbor before it has received a label for the FEC from the next-hop neighbor, the LSP is terminated at the LSR. Traffic for the destination cannot be label-switched all the way to the egress LSR. If no inner label is present, then the traffic is routed instead of switched.

In Figure 54, LSR D learns a route to some prefix. LSR D immediately maps a label for this destination and sends the label to its peers, LSR B, LSR C, LSR E, and LSR F. In the topology-driven network, the LSPs are created automatically with each peer LSR.

Figure 54: LSP Creation, Downstream-Unsolicited, Independent Control



Mapping Data

IP packets are mapped onto LSPs by one of the following methods:

- RSVP-TE tunnels can be referenced directly by static routes that you configure. You can determine which routes (routes destined for which subnets) to direct through the LSP and issue the appropriate **ip route** commands, as shown in the following example:

  host1(config-if)#**ip route 10.15.21.16 tunnel mpls:1**

  You cannot create any static routes until the tunnel interface has been created. However, the tunnel does not have to be active before you create the static routes.

- RSVP-TE tunnels are announced to IS-IS and OSPF; the IGP then uses the tunnels as next hop interfaces for its SPF calculations. For this method, you must issue the **tunnel mpls autoroute announce** command. When the LSP is established, the ingress LSR announces the LSP endpoint to the IGP. This is also referred to as *registering* the LSP. The IGP then recalculates the shortest path for all routes destined for or beyond that endpoint. You can choose to register endpoints with both IS-IS and OSPF. The following is an example registration command:

  host1(config-if)#**tunnel mpls autoroute announce isis**

- For topology-driven LSPs, LDP can modify the IP routing table to use MPLS next hops in the routing table, replacing the regular IP next hops for the corresponding routes.

- For labeled BGP routes, BGP adds routes with MPLS next hops to the appropriate VR or VRF routing table.

When IP packets arrive at the ingress LER, they are looked up in the relevant IP forwarding table and then are forwarded into an LSP. Every IP route eventually points to an IP interface. The IP interface contains IP attributes that affect how the IP packet is forwarded. IPv4 routes point only to IPv4 interfaces and IPv6 routes point only to IPv6 interfaces.

Because IP routes cannot point directly to MPLS major interfaces, MPLS automatically creates up to four dynamic IP shared interfaces that are stacked on each MPLS major interface. When you issue the **mpls** command in Interface Configuration mode, the interfaces are created dynamically and provide the interfaces an IP route needs to point to. You can specify a profile (created with the **profile** command) to configure attributes for these interfaces with the **mpls create-dynamic-interfaces** command. You can use the same command to enable or disable the creation of specific interface types or all types.

Each dynamic interface is one of the following types:

- By default, MPLS creates one dynamic IPv4 interface per MPLS major interface for non-VPN traffic. This interface is used by default for VPN traffic as well.

- By default, but only if IPv6 is enabled in the virtual router, MPLS creates one dynamic IPv6 interface per MPLS major interface for non-VPN traffic. This interface is used by default for VPN traffic as well.

- If you configure it to do so, MPLS creates one dynamic IPv4 interface per MPLS major interface for VPN traffic. If this interface is not created, then the VPN traffic uses the default IPv4 interface for non-VPN traffic.

  Typically, you request the creation of separate IPv4 interfaces for VPN traffic only when you want the IPv4 interface for VPN traffic to have different attributes, such as a different IP policy, from the IPv4 interface for non-VPN traffic. When it is acceptable for the VPN traffic and the non-VPN traffic to receive the same IP treatment, then you do not need to create separate IPv4 interfaces for the VPN traffic.

- If you configure it to do so, but only if IPv6 is enabled in the virtual router, MPLS creates one dynamic IPv6 interface per MPLS major interface for VPN traffic. If this interface is not created, then the VPN traffic uses the default IPv6 interface for non-VPN traffic.

  Typically, you request the creation of separate IPv6 interfaces for VPN traffic only when you want the IPv6 interface for VPN traffic to have different attributes, such as a different IP policy, from the IPv6 interface for non-VPN traffic. When it is acceptable for the VPN traffic and the non-VPN traffic to receive the same IP treatment, then you do not need to create separate IPv6 interfaces for the VPN traffic.

IPv6 must be enabled in the parent virtual router so that IPv6 dynamic interfaces can be created over MPLS interfaces. Otherwise, IPv6 VPNs do not work correctly,

All VPN traffic sent onto or received from the same layer 2 interface uses the same IPv4 VPN or IPv6 VPN interface. Consequently, any policy attached to the interface applies to all that VPN traffic.

IP Statistics

In the earlier architecture, the statistics for IP packets moving onto or off an LSP applied to the IP interface that was stacked on top of the LSP. Because IP interfaces are no longer stacked on LSPs in the current architecture, these statistics apply to one of the dynamic IP interfaces that is established on the same major interface the the LSP is stacked on.

However, even though the IP traffic leaving the LSP as MPLS packets is associated with a dynamic IP interface, it does not use the queue associated with that dynamic IP interface. Instead, the IP traffic uses the queues at the layer 2 interface; the MPLS major interface on which the dynamic IP interface is created does not have its own queue. As a result, queue-related statistics do not increase with the IP traffic flow on the dynamic IP interfaces. This behavior can create some confusion when you examine the output from commands such as **show egress-queue rate interface ip**.

In the following sample output, the statistics of interest are those for the layer 2 interface, atm-vc ATM9/0.10. Traffic is present as indicated by the forwarded rate value for the layer 2 interface. If no IP traffic is present, the forwarded rate for the layer 2 interface has a value of 0.

```
host1:pe1#show egress-queue rates interface atm9/0.10
                             traffic   forwarded aggregate minimum  maximum
         interface            class      rate    drop rate  rate     rate
--------------------------- ----------- --------- --------- ------- ---------
atm-vc ATM9/0.10            best-effort    116032         0 4680000 149760000
ip ATM9/0.10                best-effort         0         0 4680000 149760000
ip ip19000001.mpls.ip       best-effort         0         0 4680000 149760000
ipv6 ipv619000001.mpls.ipv6 best-effort         0         0 4680000 149760000

  Queues reported:                    4
  Queues filtered (under threshold):  0
* Queues disabled (no rate period):   0
**Queues disabled (no resources):     0
  Total queues:                       4
```

You can use the **show mpls interface brief** command to display the MPLS major interfaces. You can then view the statistics for the major interface displayed by the **show ip interface** command, as show in the following display excerpt:

```
host1:pe1#show ip interface
null0 line protocol is up, ip is up
  ...

loopback0 line protocol is up, ip is up
  ...

ATM9/0.10 line protocol Atm1483 is up, ip is up
  ...

  In Received Packets 78, Bytes 5991
    Unicast Packets 29, Bytes 2469
    Multicast Packets 49, Bytes 3522
  In Policed Packets 0, Bytes 0
  In Error Packets 0
  In Invalid Source Address Packets 0
  In Discarded Packets 0
  Out Forwarded Packets 78, Bytes 5786
    Unicast Packets 78, Bytes 5786
    Multicast Routed Packets 0, Bytes 0
  Out Scheduler Dropped Packets 0, Bytes 0
  Out Policed Packets 0, Bytes 0
  Out Discarded Packets 0
```

```
  queue 0: traffic class best-effort, bound to ip ATM9/0.10
    Queue length 0 bytes
    Forwarded packets 1, bytes 52
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

ip19000001.mpls.ip line protocol MplsMajor is up, ip is up
...
```

The **show mpls interface** command displays the queue associated with the MPLS major interface, but indicates it is bound to the layer 2 interface.

```
host1:pe1#show mpls interface
MPLS major interface ATM9/0.10
  ATM circuit type is 1483 LLC encapsulation
  Administrative state is enabled
  Operational state is up
  Operational MTU is 9180
  Received:
    1 packet
    136 bytes
    0 errors
    0 discards
    0 failed label lookups
  Sent:
    1 packet
    136 bytes
    0 errors
    0 discards

LDP information:
    10.10.10.1/24
    enabled with profile 'default'
     133 hello recv, 136 hello sent, 0 hello rej
     2 adj setup, 1 adj deleted,
        Session to 10.10.10.2 is operational (passive)
        Session statistics:
          4 label alloc, 4 label learned,
          4 accum label alloc, 4 accum label learned,
          last restart time = 00:01:49
          Rcvd: 0 notf, 8 msg, 4 mapping, 0 request
                0 abort, 0 release, 0 withdraw, 1 addr
                0 addr withdraw, 8 msgId
                0 bad mapping, 0 bad request, 0 bad abort, 0 bad release
                0 bad withdraw, 0 bad addr, 0 bad addr withdraw
                0 unknown msg type err
                last info err code = 0x00000000, 0 loop detected
          Sent: 0 notf, 8 msg, 4 mapping, 0 request
                0 abort, 0 release, 0 withdraw, 1 addr
                0 addr withdraw, 8 msgId
        Adjacency statistics:
          30 hello recv, 29 hello sent, 0 bad hello recv
          adj setup time = 00:02:19
          last hello recv time = 00:00:00, last hello sent time = 00:00:00


  queue 0: traffic class best-effort, bound to atm-vc ATM9/0.10
    Queue length 0 bytes
    Forwarded packets 1, bytes 148
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
```

MPLS Forwarding and Next-Hop Tables

An MPLS forwarding table determines how MPLS handles received MPLS packets. When an MPLS packet arrives on an MPLS major interface, MPLS looks up the outermost MPLS label of the received packet in the relevant MPLS forwarding table.

The entries in the MPLS forwarding table map labels to next hops. Each entry in the MPLS forwarding table points to an entry in the MPLS next-hop table. Each MPLS next hop points to either an interface or another MPLS next hop. The chain of MPLS next hops, which ends at an interface, informs MPLS which labels to push and where to send the MPLS packet.

For RSVP-TE tunnels, minor interfaces are created in addition to the forwarding table and next-hop table entries. One minor interface is created of each in/out segment of a tunnel. the purpose of these minor interfaces is to attach QoS and policy to an LSP.

MPLS forwarding tables consist of the following:

- One forwarding table for each MPLS virtual router. This table contains labels from the platform label space. When an MPLS packet arrives on an MPLS major interface that uses the platform label space, MPLS looks up the label in the MPLS forwarding table of the virtual router in which the major interface exists.

- One forwarding table for each MPLS major interface that uses the interface label space. This table contains labels from the interface label space of that major interface. When an MPLS packet arrives on an MPLS major interface that uses the interface label space, MPLS looks up the label in the MPLS forwarding table for that particular major interface.

The signaling protocols add entries to the MPLS forwarding tables. You cannot manually create an MPLS forwarding entry. The signaling protocols set the following attributes for each entry placed in the forwarding table:

- An MPLS in label that is matched against the outermost label of the received MPLS packet.

- The MPLS next hop, which specifies the actions to be performed on the MPLS packet. MPLS next hops can be chained together to create complex actions.

- A spoof check field that specifies the type of spoof checking is performed to determine whether the MPLS packet arrived from a legitimate source. See *Spoof Checking* on page 213 for more information.

You can enable statistics collection for MPLS forwarding table entries. See *Monitoring MPLS* on page 318 for more information.

Spoof Checking

The MPLS forwarding table enables MPLS to determine whether an MPLS packet received from an upstream neighbor contains an MPLS label that was advertised to that neighbor. If not, the packet is dropped. Each entry in the forwarding table has a spoof check field that specifies the type of checking that must be performed for the associated in label. The signaling protocol (BGP, LDP, or RSVP) that populates the entry in the MPLS forwarding table sets the spoof check field.

MPLS supports the following types of spoof checking:

- Router spoof checking—MPLS packets are accepted only if they arrive on an MPLS major interface that is in the same virtual router as the MPLS forwarding table.
- Interface spoof checking—MPLS packets are accepted only if they arrive on the particular MPLS major interface identified in the spoof check field.

You can use the **show mpls forwarding** command to view the spoof check field for an MPLS forwarding table entry.

IP and IPv6 Tunnel Routing Tables

The IP and IPv6 tunnel routing tables contain routes that point only to tunnels, such as MPLS tunnels. The tunnel routing table is not used for forwarding. Instead, protocols resolve MPLS next hops by looking up the routes in the table. For example, BGP uses the table to resolve indirect next hops for labeled routes.

BGP, LDP, and RSVP-TE can contribute routes to the table. LDP adds all destinations that can be reached by means of labels learned from downstream LDP neighbors. RSVP-TE adds only MPLS tunnel endpoints. BGP also adds routes to the tunnel table in certain cases. Routes added by any of these protocols include the effective metric.

For example, in a BGP/MPLS VPN topology, LDP or RSVP-TE adds routes to the tunnel routing table for all available tunnels. BGP performs a lookup in the tunnel routing table so that it can resolve indirect next hops.

You can use the **clear ip tunnel-routes** or **clear ipv6 tunnel-routes** command to clear routes from the IP tunnel routing table and notify all protocols to add their routes again. You might do this, for example, to reapply routing policies when the policies are changed.

clear ip tunnel-routes

- Use to clear and then refresh a specified IPv4 dynamic route or all IPv4 dynamic routes from the tunnel routing table of the virtual router or a specified VRF.
- This command takes effect immediately.
- Example

  host1(config)#**clear ip tunnel-routes ***

- There is no **no** version.

clear ipv6 tunnel-routes

- Use to clear and then refresh a specified IPv6 dynamic route or all IPv6 dynamic routes from the tunnel routing table of the virtual router or a specified VRF.
- This command takes effect immediately.
- Example

  host1(config)#**clear ipv6 tunnel-routes ***

- There is no **no** version.

MPLS Interfaces and Interface Stacking

The JUNOSe implementation of MPLS employs MPLS major, minor, and shim interfaces.

MPLS Major Interfaces

An MPLS major interface must be stacked on a layer 2 interface to send or receive MPLS packets on that interface. Each MPLS major interface exists in a particular virtual router.

MPLS major interfaces can use the platform label space or the interface label space. Which type of label space is used by the major interface is determined by the layer 2 interface on which the major interface is stacked. If the layer 2 interface is an ATM AAL5 interface, the major interface uses the interface label space. For all other layer 2 interface types, the major interface uses the platform label space.

When an MPLS packet arrives on the MPLS major interface, MPLS looks up the label of the received MPLS packet, the in label, in the MPLS forwarding table that is associated with the major interface. For major interfaces using the platform label space, the lookup is in the MPLS forwarding table of the VR. For major interfaces using the interface label space, the lookup is in the MPLS forwarding table of the major interface.

You use the **mpls** command in Interface Configuration mode to create or remove MPLS major interfaces. Some other commands create an MPLS major interface if it does not already exist.

You can configure the following attributes for each MPLS major interface:

- The administrative state, enabled or disabled, configured with the **mpls disable** command.
- The range of ATM VPIs used as interface labels for major interfaces stacked on ATM AAL5 interfaces, configured with the **mpls atm vpi range** command.
- The range of ATM VCIs used as interface labels for major interfaces stacked on ATM AAL5 interfaces, configured with the **mpls atm vci range** command.

MPLS Minor Interfaces

When you configure an LSP with the **interface tunnel mpls** command, RSVP-TE creates an MPLS minor interface to represent the head of the LSP. MPLS minor interfaces are also created by RSVP-TE on the transit and tail LSRs when the LSP is signaled. Only RSVP-TE creates MPLS minor interfaces. Neither BGP nor LDP create them.

These minor interfaces are used to associate policy or a QoS profile with an LSP (either on an LSR or an LER). This minor interface is created automatically by the signaling protocol. Minor interfaces are not saved in NVS. Use the **show mpls interface minor** command to view the minor interfaces.

The following attributes of the minor interface are set by RSVP-TE:

- The UID of the minor interface, assigned automatically when the interface is created.
- The operational state of the interface, up or down.
- Whether the interface is an ingress MPLS minor interface used to receive traffic or an egress MPLS minor interface used to transmit traffic.

MPLS Shim Interfaces

MPLS shim interfaces are stacked on layer 2 interfaces to provide layer 2 services over MPLS or to create local cross-connects by cross-connecting the layer 2 interface to another layer 2 interface. For more information about MPLS shim interfaces, see *JUNOSe BGP and MPLS Configuration Guide, Chapter 5, Configuring Layer 2 Services over MPLS*.

Interface Stacking

MPLS interface stacking differs depending on whether the platform label space (Figure 55) or the interface label space (Figure 56) is used.

Figure 55: MPLS Interface Stacking for the Platform Label Space

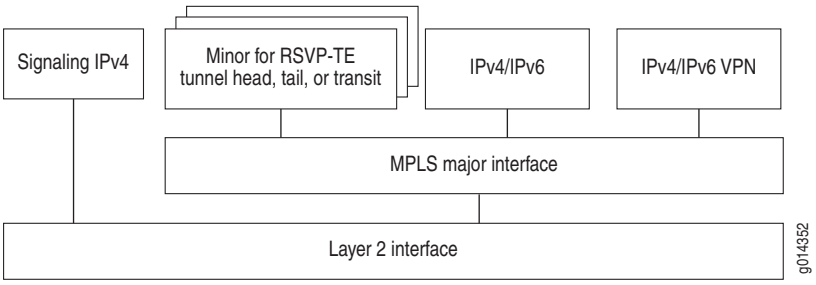
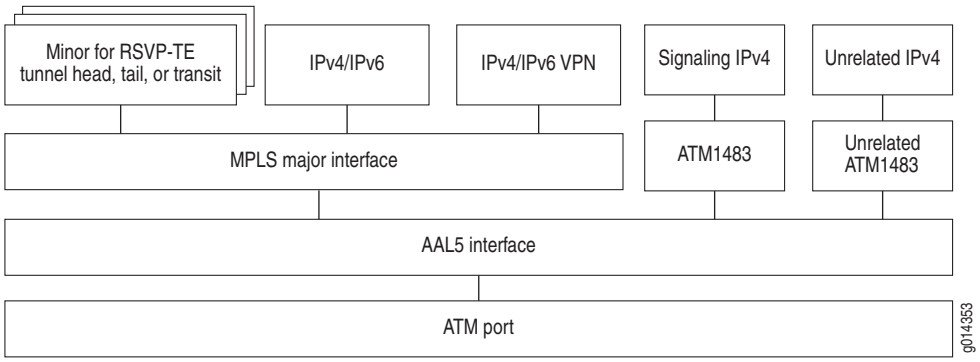


Figure 56: MPLS Interface Stacking for the Interface Label Space



Label Distribution Protocols

Label distribution protocols create and maintain the label-to-FEC bindings along an LSP from MPLS domain ingress to MPLS domain egress. A label distribution protocol is a set of procedures by which one LSR informs a peer LSR of the meaning of the labels used to forward traffic between them. It enables each peer to learn about the other peer's label mappings. The label distribution protocol provides the information MPLS uses to create the forwarding tables in each LSR in the MPLS domain.

NOTE: Label distribution protocols are sometimes referred to as signaling protocols. However, label distribution is a more accurate description of their function and is preferred in this text.

The following protocols are currently used for label distribution:

- BGP—Border Gateway Protocol
- LDP—Label Distribution Protocol
- RSVP-TE—Resource Reservation Protocol with traffic-engineering extensions that enable label binding and explicit route capability

NOTE: To reduce confusion, this text uses the lowercase term, label distribution protocol, to refer to the generic class of protocols. The acronym, LDP, refers only to the particular protocol named Label Distribution Protocol.

BGP and LDP have no traffic-engineering capability and support only best-effort LSPs. LDP supports topology-driven MPLS networks in best-effort, hop-by-hop implementations. RSVP-TE is used primarily for MPLS applications that require traffic engineering (TE) or quality of service (QoS) capabilities, but they also support best-effort LSPs.

LDP Messages and Sessions

LDP creates reliable sessions by running over TCP. You do not have to explicitly configure LDP peers, because each LSR actively discovers all other LSRs to which it is directly connected. LDP is a *hard-state* protocol, meaning that after the LSP is established, it is assumed to remain in place until it has been explicitly torn down. This is in contrast to RSVP-TE, which is a *soft-state* protocol. See *RSVP-TE Messages and Sessions* on page 219.

LDP uses many messages to create LSPs, classified in the following four types:

- Discovery—To identify other LSRs
- Adjacency—To create, maintain, and end sessions between LSRs
- Label advertisement—To request, map, withdraw, and release labels
- Notification—To provide advisory and error information

Unlike the other LDP messages, the discovery process runs over UDP. Each LSR periodically broadcasts a link hello message to the well-known UDP port, 646. Each LSR listens on this port for link hello messages from other LSRs. In this manner, each LSR learns about all other LSRs to which it is directly connected, creating link hello adjacencies. When an LSR learns about another LSR, it establishes a TCP connection to the peer on well-known TCP port 646 and creates an LDP session on top of the TCP connection.

A transport address for the local peer is advertised in LDP discovery hello messages. Interfaces that use the platform label space default to the LSR router ID for the transport address. You can use the **mpls ldp discovery transport-address** command to specify an arbitrary IP address as the transport address.

LDP can also discover peers that are not directly connected if you provide the LSR with the IP address of one or more peers by means of an access list. The LSR sends targeted hello messages to UDP port 646 on each remote peer. If the targeted peer responds with a targeted hello message to the initiator, a targeted hello adjacency is created and session establishment can proceed.

In certain cases, a targeted hello adjacency to directly connected peers might be useful. If an LSR receives both a link hello message and a targeted hello message from the same initiator, only a single LDP session is established between the LSRs.

By default, because all LSRs listen on the well-known port, they all attempt to create a session with the originator. You can use the **mpls ldp link-hello disable** command to suppress the transmission of link hello messages. Thereafter, sessions are formed only with peers contacted with targeted hello messages.

The LDP peers exchange session initialization messages that include timer values and graceful-restart parameters. An LSR responds with a keepalive message if the values in the initialization message are acceptable. If any value is not acceptable, the LSR responds instead with an error notification message, terminating the session. After a session is established, LDP peers exchange keepalive messages that verify continued functioning of the LSR. Failure to receive an expected keepalive message causes an LSR to terminate the LDP session.

Label mapping and distribution use downstream-unsolicited, independent control.

With downstream-unsolicited, independent control, an LSR creates a label binding whenever it learns a new IGP route; the LSR sends a label mapping message immediately to all of its peer LSRs—upstream and downstream—without having received a label request message from any peer. The LSR sends the label mapping message regardless of whether it has received a label mapping message from a downstream LSR. This is the label distribution method employed in a topology-driven MPLS network.

A downstream LSR can send a label withdrawal message to recall a label that it previously mapped. If an LSR that has received a label mapping subsequently determines that it no longer needs that label, it can send a label release message that frees the label for use.

RSVP-TE Messages and Sessions

RSVP is described in RFC 2205. Multiple RFCs enable extensions to RSVP for traffic engineering. The router supports the extended version of RSVP, referred to as RSVP-TE.

RSVP-TE is "unreliable" because it does not use TCP to exchange messages. In contrast to LDP—a hard-state protocol—RSVP-TE is a *soft-state* protocol, meaning that much of the session information is embedded in a state machine on each LSR. The state machine must be refreshed periodically to avoid session termination. LSRs send path messages to downstream peers to create and refresh local path states. LSRs send resv messages to upstream peers in response to path messages to create and refresh local resv states. A session is ended if the state machine is not refreshed within the RSVP tunnel timeout period, which is determined as follows:

$$\text{RSVP tunnel timeout period in seconds} = \left\{ [(\text{cleanup timeout factor} + 0.5) \times 1.5] \times \left(\frac{\text{refresh period}}{1000} \right) \right\}$$

For example, for the default values,

$$\text{RSVP tunnel timeout period in seconds} = \left\{ [(3 + 0.5) \times 1.5] \times \left(\frac{30{,}000}{1000} \right) \right\} = 157.5$$

RSVP-TE messages carry *objects* consisting of type-length-values (TLVs). The *label request object* instructs the endpoint LSR to return an resv message to establish the LSP. The resv message contains the *label object*, the label used for the FEC. Both the path and resv messages carry the *record route object*, which records the route traversed by the message.

An upstream LSR sends a pathtear message when its path state times out as a result of not being refreshed. The pathtear message removes the path and resv states in each LSR as it proceeds downstream. Downstream LSRs similarly send the resvtear message when their resv state times out to remove the resv states in upstream LSRs.

If a downstream LSR determines that it received an erroneous path message, it sends a patherr message to the sender. If a reservation (label) request fails, the request initiator sends a resverr message to the downstream LSRs. Both of these messages are advisory and do not alter path or resv state.

RSVP-TE State Refresh and Reliability

RSVP-TE employs refresh messages to synchronize state between neighbors and to recover from lost RSVP-TE messages of other types. RSVP-TE by default does not provide for reliable transmission. Each node is responsible for the transmission of RSVP-TE messages to its neighbors and relies on periodic state refreshes to recover from lost messages.

RSVP-TE refresh reduction provides a means to increase reliability while reducing message handling and synchronization overhead. Issuing the **mpls rsvp refresh-reduction** command enables the following features:

- The message ID object is included in RSVP-TE messages to provide a unique message ID on a per-hop basis. In refresh messages, it indicates to the receiving node that the message is a refresh message, eliminating the need for the node to completely analyze the message and thus reducing the processing overhead at the receiver.

- RSVP-TE uses a message acknowledgment mechanism to support reliable message delivery on a per-hop basis. Messages that include the message ID object also include a request for acknowledgment. Upon receipt, the receiving node returns a message ack object, enabling the sending node to determine whether a message was lost and triggering a retransmission as necessary.

- Summary refresh (srefresh) messages refresh the state previously advertised in path or resv messages that included message ID objects. The srefresh message carries the unique message ID as state identifier, eliminating the need to send whole refresh messages and reducing the overhead needed to maintain RSVP-TE state synchronization. This method maintains RSVP-TE's ability to indicate when the state is lost and to adjust to changes in routing. The srefresh message can carry message IDs for multiple RSVP-TE sessions.

Issuing the **mpls rsvp message-bundling** command enables RSVP-TE to use bundle messages, each of which includes multiple standard RSVP-TE messages, to reduce the overall message processing overhead.

BGP Signaling

You can use BGP as a label distribution protocol both for IP routes and for VPN routes.

When BGP distributes a particular IP route, it can also distribute an MPLS label that has been mapped to that route, as described in RFC 3107. The MP-BGP extensions (RFC 2858) enable the BGP update message to carry both the route and the label mapping information for the route. The label is encoded into the NLRI field of the attribute, and the SAFI field is set to 4 to indicate that the NLRI contains a label. A BGP speaker can use BGP to send labels to a particular BGP peer only if that peer advertised the capability to process update messages with SAFI 4. BGP speakers advertise this capability only to peers for which the **neighbor send-label** command has been configured.

When BGP advertises labeled routes, it adds a label-to-next-hop mapping (cross-connect) to the MPLS forwarding table. This mapping consists of the in label that BGP allocates from the platform label space plus the MPLS next hop information related to the labeled route's next hop.

BGP can also distribute labels for VPN routes in BGP/MPLS VPNs. In a BGP/MPLS VPN network, BGP is used to exchange the routes of a particular VPN among the provider edge routers attached to the VPN. To ensure that routes from different VPNS remain distinct even if the VPNs use overlapping address spaces, an MPLS label is assigned to each route within the VPN. BGP distributes both the VPN routes and the associated MPLS label for each route.

The label mapping information for a particular VPN route is included in the same BGP update message that distributes the route. The label is encoded into the NLRI field of the attribute, and the SAFI field has a value of 128 to indicate that the NLRI contains both an RD (route distinguisher) and a label.

For more information on BGP capabilities, see *Chapter 1, Configuring BGP Routing*. For more information on MP-BGP extensions, NLRIs, and BGP/MPLS VPNs, see *Chapter 3, Configuring BGP-MPLS Applications*.

ECMP

MPLS supports equal-cost multipath (ECMP) labels. A maximum of 16 MPLS paths is supported; 4 paths are available by default. On LERs, MPLS ECMP next hops can be used in the IP routing table for non-VPN and VPN routes. On LSRs, an incoming label can point to either an MPLS ECMP next hop or an IP ECMP.

The signaling protocol determines whether ECMP next hops are used. For example, LDP can learn multiple labels for a route from different downstream peers (or one label from a downstream peer that has parallel connections to the router). LDP then creates an MPLS ECMP next hop that can be used in the IP routing table. If LDP also advertises a label, then a forwarding entry is added to the MPLS forwarding table with the ECMP next hop.

LDP Discovery Mechanisms

LDP uses two different mechanisms for peer discovery. Peer discovery removes the need to explicitly configure the label-switching peers for an LSR.

- LDP uses the basic discovery mechanism to discover directly connected LDP peers.
- LDP uses the extended discovery mechanism to discover peers that are not directly connected.

Basic Discovery Mechanism

To discover directly connected peers, LSRs periodically send out LDP link hellos on the interface. The link hellos are contained in UDP packets that are addressed to the well-known LDP discovery port, 646. The destination address for the ports is 224.0.0.2. Using this port and address ensures that the hellos are sent to all routers on the interface's subnet.

The link hello includes the LDP identifier for the label space that the LSR intends to use for the interface. In the JUNOSe implementation, this is always the platform label space, so the LDP identifier specifies the LSR ID and a value of 0 for the label space. The link hello also includes other information, such as the hello hold time configured on the interface. The hello hold time specifies how long an LSR maintains a record of hellos received from potential peers.

When an LSR receives a link hello, it identifies the sending LSR as a potential LDP peer on that interface. The LSRs form a hello adjacency to keep track of each other.

The basic discovery mechanism is enabled by default when you enable LDP on an interface. You can configure the link hellos in the LDP profile with the **hello hold-time** and **hello interval** commands. You can configure a transport IP address to be globally included in link hellos with the **mpls ldp discovery transport-address** command.

Extended Discovery Mechanism

To discover LDP peers that are not directly connected, LSRs periodically send out LDP targeted hellos to potential peers. The targeted hellos are contained in UDP packets that are addressed to the well-known LDP discovery port, 646. The destination address for the ports is a specific targeted address. LDP sends targeted hellos when you configure one or more IP addresses in a targeted-hello send list. In a layer 2 Martini circuit, targeted hellos are automatically sent to the remote PE neighbor (the base tunnel endpoint). See *JUNOSe BGP and MPLS Configuration Guide, Chapter 5, Configuring Layer 2 Services over MPLS* for information about layer 2 circuits.

The targeted hello includes the LDP identifier for the label space that the LSR intends to use. In the JUNOSe implementation, this is always the platform label space, so the LDP identifier specifies the LSR ID and a value of 0 for the label space. The targeted hello also includes other information, such as the targeted-hello hold time, which is configured globally. The targeted-hello hold time configures how long an LSR waits for another targeted hello from its peer before declaring the adjacency to be down.

Unlike basic discovery, where hellos are sent by all LSRs, extended discovery is initiated by one LSR that targets a specific LSR. The initiating LSR periodically sends targeted hellos to the targeted LSR. The targeted LSR then determines whether to respond to the targeted hello or to ignore it. If the targeted LSR responds to the sender, it does so by periodically sending targeted hellos to the initiating LSR. The exchange of targeted hellos constitutes a hello adjacency for the two LSRs.

Targeted hello values are configured globally with the **mpls ldp targeted-hello holdtime**, **mpls ldp targeted-hello interval**, **mpls ldp targeted-hello receive list**, and **mpls ldp targeted hello send list** commands.

Traffic Engineering

MPLS traffic engineering (TE) is the ability to establish LSPs according to particular criteria (*constraints*) in order to meet specific traffic requirements rather than relying on the path chosen by the conventional IGP. The constraint-based IGP examines the available network resources and calculates the shortest path for a particular tunnel that has the resources required by that tunnel. Traffic engineering enables you to make the best use of your network resources by reducing overuse and underuse of certain links.

Constraint-based routing (CR) makes traffic engineering possible by considering resource requirements and resource availability rather than merely the shortest path calculations. Constraints are determined at the edge of the network and include criteria such as required values for bandwidth or required explicit paths. You can use RSVP-TE as the label distribution protocol for traffic engineering. The IGP propagates resource information throughout its network. RSVP-TE employs downstream-on-demand, ordered control for label mapping and distribution.

Explicit routing specifies a list or group of nodes (hops) that must be used in setting up the tunnels. CR explicit paths can be *strict* or *loose*. Strict paths specify an exact physical path, including every physical node. Loose paths include hops that have local flexibility; the hop can be a traditional interface, an autonomous system, or an LSP.

LSP Backup

You can configure multiple LSPs to the same destination. By configuring different tunnel metrics for these LSPs, you can force a ranking or priority of use for the LSPs. In this scenario, all the configured LSPs are up and active. If the LSP in use develops problems and goes down, traffic is diverted to the LSP having the next best metric.

Path Option

You can configure multiple paths for an LSP with the **tunnel mpls path-option** command. Each path option has an identifying number; the lower the number the higher the preference for that path option. In this scenario, only a single LSP is up and active at a time. If the path option currently in use by an LSP goes down, MPLS tries to reroute the tunnel using the path option with the next highest preference. In certain circumstances—for example, when a tunnel is preempted by another—MPLS first attempts to reroute the tunnel with the current path option.

Reoptimization

You can use the traffic-engineering reoptimization capability to ensure that the best path is being used. Suppose the current path goes down and MPLS switches to an alternate path that is not as good as the failed path. You can have MPLS periodically search—according to a specified schedule—for a path better than the alternate by configuring the reoptimization timer. For example, you might configure MPLS to search for a better path every 10 minutes; if it finds a better path, it switches.

On the other hand, you might be concerned about route flapping. If a path goes down and then comes back up, perhaps it will continue to do so. In this case, you might not ever want to go back to a path that goes down. To accomplish this, you can configure reoptimization to never occur.

When you do not want the initial path to change—that is, when you want to pin the route—you can disable reoptimization globally by setting the timer to 0. Alternatively, you can disable reoptimization on a per-tunnel basis by using the **lockdown** option with the **tunnel mpls path-option** command. LSP paths are always pinned until the next reoptimization.

Finally, you can manually force an immediate reoptimization. See *Global Configuration Tasks* on page 228 for information about configuring reoptimization.

Configuring Tunnels

You can use either of the following methods to configure RSVP-TE tunnels:

- Configure individual tunnels with the **interface tunnel mpls:***tunnelName* command.
- Configure multiple tunnels with the same set of parameters with the **mpls tunnels profile** command.

Tracking Resources for Traffic Engineering

MPLS traffic engineering uses *admission control* to keep track of resource information. Admission control has an accounting feature that ensures that requests are not accepted when the router does not have sufficient resources to accommodate them.

Currently, bandwidth (BW) and bandwidth-related information are the only resources tracked and used for traffic engineering. Admission control determines whether a setup request can be honored for an MPLS LSP with traffic parameters.

Admission control provides bandwidth information to the IGP protocols, ISIS and OSPF. As new LSPs are created, the available bandwidth decreases. The IGPs can subsequently advertise this information and use it for SPF calculations to determine paths that satisfy the traffic requirements. You can configure readvertisement to occur periodically or when the change crosses some threshold.

Starting Admission Control

Admission control operates on a router-wide basis rather than a per-virtual-router basis. Admission control of resources begins when either of the following occurs:

- You configure resource-related information about an interface, including bandwidth (either total bandwidth or MPLS reservable bandwidth), flooding frequency, flooding threshold, administrative weight, or attribute flags.
- MPLS begins to use admission control services; for example, by attempting to set up a constraint-based LSP.

Admission Control Interface Table

Configuring bandwidth on an interface creates an entry for the interface in the admission control interface table. Each entry in the table stores the following information per interface:

- Maximum (physical or line-rate) bandwidth
- Maximum reservable bandwidth
- The following information per IP class (currently a single, default class)
  - Total available (unreserved) bandwidth
  - Available bandwidth at each MPLS priority level
- Resource flooding threshold and period

The resource flooding threshold and period together control the flooding of the resource information by the IGP protocols, IS-IS and OSPF.

Configuring Traffic-Engineering Resources

You can configure the following resource-related information about an MPLS interface (at either the major interface or subinterface level):

- Bandwidth—Total bandwidth that can be reserved on the interface
- Flooding thresholds—Sets of absolute percentages of total reservable bandwidth that trigger the new bandwidth value to be flooded throughout the network; flooding is triggered when bandwidth increases past any up threshold value or decreases past any down threshold value
- Flooding frequency—Periodicity with which the bandwidth value is flooded, apart from any flooding due to value changes
- Administrative weight—Weight assigned to the interface that supersedes any assigned by the IGP
- Attribute flags—32-bit value that assigns the interface to a resource class and enables a tunnel to discriminate among interfaces by matching against tunnel affinity bits

Monitoring Resources

See *Monitoring MPLS* on page 318 for information about displaying information related to traffic-engineering resources.

LSP Preemption

You can develop a preemption strategy whereby a new LSP can claim resources from an existing LSP. Each tunnel can be configured with a *setup* priority and a *hold* priority. Priority levels range from 0 (highest priority) through 7 (lowest priority).

If traffic engineering admission control determines that there are insufficient resources to accept a request to set up a new LSP, the setup priority is evaluated against the hold priority of existing LSPs. An LSP with a hold priority lower than the setup priority of the new LSP can be preempted. The existing LSP is terminated to make room (free resources) for the new LSP. You must assign priorities according to network policies to prevent resource poaching and LSP thrashing.

Topology-Driven LSPs

Topology-driven LSPs are implemented for best-effort, hop-by-hop routing. In topology-driven LSP mode, LDP automatically sets up LSPs for IGP, direct, and static routes, subject to filtering by access-lists. JUNOSe supports downstream-unsolicited LDP using the platform label space.

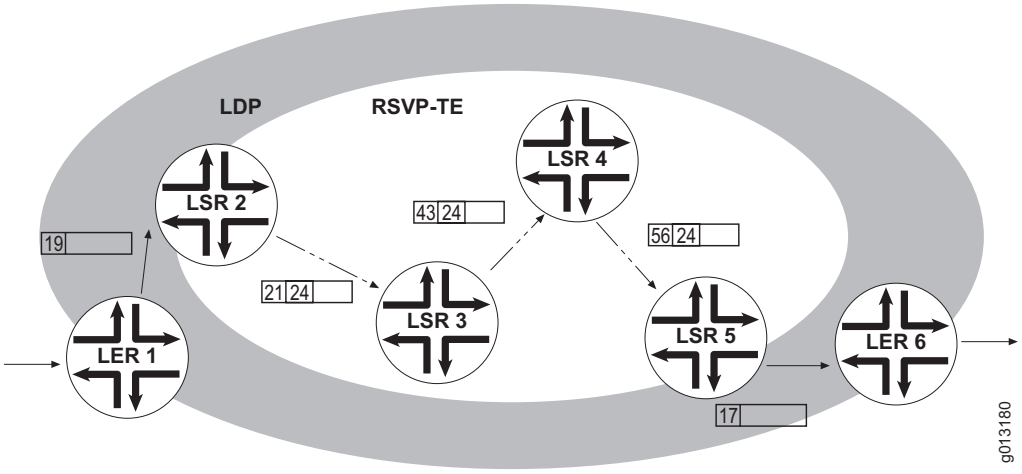
If you use the topology-driven LSP mode to forward plain IP packets, use the **ldp ip-forwarding** command to place LSPs into the IP routing table for forwarding plain IP traffic.

You can use the **mpls ldp advertise-labels** command to limit the number of routes for which labels are advertised. In most cases, you can issue **mpls ldp advertise-labels host-only**.

LDP over RSVP-TE

If you are running RSVP-TE in the core, LDP can tunnel through the core by stacking an LDP LSP over an RSVP-TE LSP, as shown in Figure 57. With LDP over RSVP-TE, LDP establishes targeted sessions among the LDP routers at the edge of the RSVP core. From the perspective of the LDP LSP, the RSVP-TE core is a single hop.

Figure 57: LDP Tunneled Through an RSVP-TE Core



In the network topology illustrated in Figure 57, the RSVP-TE LSP consists of LSR 2, LSR 3, LSR 4, and LSR 5. The LDP LSP consists of LER 1, LSR 2, LSR 5, and LER 6. The RSVP-TE tunnel appears to LDP as a single hop.

The initial LDP label 19 is switched with label 24 at LSR 2. Because this is the entrance to the RSVP-TE tunnel, label 21 is pushed onto the stack. Label 21 is switched with label 43 at LSR 3. Label 43 is switched with label 56 at LSR 4. LSR 5 pops both labels, pushes label 17, and forwards the packet to LER 6.

On the LDP routers that are on the edge of the core, you must configure a list of peer addresses. The LDP router sends targeted hello messages to those addresses in order to establish targeted sessions across the RSVP-TE domain. The list includes other LDP routers on the edge of the core; for example, in Figure 57, you include the address of LSR 5 in the list configured on LSR 2.

Configuration Tasks

Configuring an MPLS network consists of the following sets of tasks:

- Global tasks—Perform these tasks to configure settings common to all MPLS usage on a given LSR.
- (Optional) Interface profile tasks—Perform these tasks to configure LDP or RSVP-TE interface profiles.
- Interface tasks—Perform these tasks to configure each interface on an LSR that uses MPLS.
- Tunnel or topology-driven LSP tasks—Perform these tasks to configure MPLS tunnels or topology-driven LSPs.
- (Optional) Tunnel profile tasks—Perform these tasks to configure a profile that contains settings to be used by multiple MPLS tunnels.

In this section, pure MPLS configuration tasks are distinguished from LDP and RSVP-TE configuration tasks.

Many users find it convenient to configure MPLS by completing the tasks in each category before moving to the next category. However, you do not have to complete the tasks in the listed order.

The type of network you want to implement determines which sets of tasks you must complete, as indicated in Table 25.

Table 25: Configuration Tasks by Type of Network

Task Set	Traffic Engineering Network	Topology-Driven Network (Best-Effort, Hop-by-Hop, LDP)
Global	Yes	Yes
Interface Profile	Optional	Optional
Interface	Yes	Yes
Tunnel	Yes	No
Tunnel Profile	Yes	No

While the basic configuration tasks are covered in this section, additional configuration tasks are described in later sections of this chapter.

Global Configuration Tasks

Complete these tasks to configure a virtual router as an LSR. You perform these commands in Global Configuration mode. The following sequence is arbitrary; you can perform these tasks in any order.

After you have completed the global configuration tasks, proceed to *LDP Interface Profile Configuration Tasks and Command* on page 242.

MPLS Tasks and Commands

In a typical network, you perform only the first task. You might also perform the optional configuration tasks, but typically do not need to do so.

1. Enable MPLS on a virtual router.

   host1(config)#**mpls**

2. (Optional) Configure the time-to-live field placed in the MPLS header when a label is first added to an IP packet.

   host1(config)#**mpls ip propagate-ttl forwarded**

3. (Optional) Configure the tunneling model for differentiated services. See *MPLS and Differentiated Services* on page 300 for more information and command descriptions.

   host1(config)#**mpls tunnel-model uniform**

4. (Optional) Specify whether to use the TOS value or the UPC value of the packet as the value of the EXP bits when the router acts as an LER. See *MPLS and Differentiated Services* on page 300 for more information and command descriptions.

   host1(config)#**mpls copy-upc-to-exp**

5. (Optional) Specify whether the EXP bits for VPN MPLS labels can be modified by EXP bit mapping or by policy for differentiated services. See *MPLS and Differentiated Services* on page 300 for more information and command descriptions.

   host1(config)#**mpls preserve-vpn-exp**

6. (Optional) Specify whether to create dynamic IP interfaces on top of MPLS major interfaces and optionally what profile to use for them.

   host1(config)#**mpls create-dynamic-interfaces ip on-major-interfaces**

7. (Optional) Remove and then reestablish existing dynamic IPv4 and IPv6 interfaces on top of MPLS major interfaces.

   host1#**clear mpls dynamic-interfaces on-major-interfaces**

clear mpls dynamic-interfaces on-major-interfaces

- Use to remove and re-create dynamic IPv4 interfaces and dynamic IPv6 interfaces from MPLS major interfaces.
- Use the **ip** keyword to specify that only IPv4 interfaces are removed and recreated. Use the **ipv6** keyword to specify that only IPv6 interfaces are removed and recreated. When you do not include either of these keywords, both IPv4 and IPv6 dynamic interfaces are removed and recreated.
- You can specify an interface type and specifier from which the dynamic interfaces are removed. Otherwise, the dynamic interfaces are removed from all MPLS major interfaces.
- You can use this command to reapply policies related to dynamic IPv4 and IPv6 interfaces on top of MPLS major interfaces.
- Example

  host1#**clear mpls dynamic-interfaces ipv6 on-major-interfaces**

- There is no **no** version.

mpls

- Use to create, enable, disable, or delete MPLS on a virtual router.
- MPLS does not exist by default and must be created, either explicitly with this command, or implicitly with another **mpls** configuration command. If you create MPLS implicitly, for example by issuing the command **mpls lsp retries 10**, MPLS remains disabled until you enable it, by issuing the **mpls** command.
- Example

  host1(config)#**mpls**

- Use the **no** version to halt MPLS on the virtual router and delete the MPLS configuration.

mpls create-dynamic-interfaces

- Use to specify whether dynamic IP interfaces are created on top of MPLS major interfaces, and what profile to use to create them.
- By default, one IPv4 dynamic interface without a profile is created and used for both VPN and non-VPN traffic. If IPv6 is enabled on the virtual router, then by default, one IPv6 dynamic interface without a profile is created and used for both VPN and non-VPN traffic.
- Example

  host1(config)#**mpls create-dynamic-interfaces ip on-major-interfaces profile v4intf**

- Use the **no** version to restore the default behavior.

mpls ip propagate-ttl

- Use to configure the time-to-live field placed in the MPLS header when a label is first added to an IP packet.
- This command is enabled by default, causing the TTL field to be copied from the IP packet header and enabling the **traceroute** command to show all the hops in the network.
- This command applies only to the tunnel ingress.
- Example

  host1(config)#**mpls ip propagate-ttl**

- Use the **default** version to revert to the global default, causing the TTL field to be copied from the IP packet header and enabling the **traceroute** command to show all the hops in the network.
- Use the **no** version to specify a fixed TTL value of 255 to hide the structure of the MPLS network (all packets) from all traffic, preventing a **traceroute** command from discovering and displaying LSP hops. The **no mpls ip propagate-ttl** command changes the TTL processing tunnel model from the default uniform model to the pipe model.

  You can specify the types of packets to be hidden from **traceroute** by using the following keywords with the **no** version:

  - **forwarded**—**traceroute** cannot show the hops for forwarded packets; this most common application of the command enables you to hide the structure of the MPLS network from your customers. Sets the tunnel model to pipe for forwarded packets.

    host1(config)#**no mpls ip propagate-ttl forwarded**

  - **local**—**traceroute** cannot show the hops for local packets. Sets the tunnel model to pipe for locally originated packets.

    host1(config)#**no mpls ip propagate-ttl local**

  You can subsequently specify these keywords with the affirmative version of the command to stop hiding the packets from **traceroute**.

LDP Tasks and Commands

Typically, you do not configure anything for LDP at the global level, but you can perform the following optional tasks.

1. (Optional) Enable LDP and topology-driven LSP. Any LDP-related command creates LDP implicitly, negating the need to issue this command.

   host1(config)#**mpls ldp**

2. (Optional) Configure the redistribution of IGP routes to LDP.

   host1(config)#**mpls ldp redistribute ospf route-map boston5**

3. (Optional) Configure lists of peer addresses that targeted hello messages are sent to or accepted from.

   host1(config)#**mpls ldp targeted-hello send list 10.21.5.87**
   host1(config)#**mpls ldp targeted-hello receive list 192.168.45.25**

4. (Optional) Configure the hold time and interval values for targeted hello messages used in LDP extended discovery.

   host1(config)#**mpls ldp targeted-hello holdtime 90**
   host1(config)#**mpls ldp targeted-hello interval 30**

5. (Optional) Configure LDP session retry values.

   host1(config)#**mpls ldp session retry-time 2**
   host1(config)#**mpls ldp session retries 1800**

6. (Optional) Configure the period that LDP negotiates with its peer for which the LDP session is maintained in the absence of any LDP messages.

   host1(config)#**mpls ldp session holdtime 1800**

7. (Optional) Configure the interval at which LDP sends session keepalive messages.

   host1(config)#**mpls ldp session keepalive-time 180**

8. (Optional) Specify an IP address to be advertised to peers as the transport address in discovery hello messages.

   host1(config)#**mpls ldp discovery transport-address 192.168.34.2**

9. (Optional) Configure independent control as the method of label distribution that LDP uses.

   host1(config)#**mpls ldp independent-control**

10. (Optional) Configure LDP to advertise the explicit null label or a non-null label for the egress router to achieve ultimate hop popping.

    host1(config)#**mpls ldp egress-label explicit-null**

For topology-driven LSPs, perform the following LDP configuration tasks.

1. (Optional) Configure the LSR to create topology-driven LSPs. Enabling LDP automatically creates topology-driven LSPs.

   host1(config)#**mpls topology-driven-lsp**

2. (Optional) Specify filters for the routes and peers to which the labels are advertised.

   host1(config)#**mpls ldp advertise-labels host-only**

3. (Optional) Specify the LSPs to be put into the IP routing table for forwarding plain IP traffic.

NOTE: This step is not optional if you are using a topology-driven network to forward plain IP packets.

   host1(config)#**ldp ip-forwarding host-only**

4. (Optional) Establish a policy governing the distribution of incoming LDP labels.

   host1(config)#**mpls ldp advertise-labels for boston1**

5. (Optional) Remove and then reestablish existing LDP LSPs.

   host1#**clear mpls ldp**

clear mpls ldp

- Use to remove and then reestablish all existing LDP LSPs; this action affects data traffic on an LSP that is in use when you issue the command.
- You can clear all LDP LSPs, or only those that are established to a particular prefix or neighbor.
- Use this command when you when you want to restart topology-driven LDP.
- Use this command when you have modified or created policies or access lists (with the **mpls ldp-ip-forwarding** and **mpls ldp advertise-labels** commands) and want them to be applied to LDP LSPs that are already in an up state.
- Example

  host1#**clear mpls ldp**

- There is no **no** version.

mpls ldp

- Use to enable LDP and automatically enable topology-driven LSPs.
- Because any LDP-related command creates LDP implicitly, you do not need to issue this command.

- Example

  host1(config)#**mpls ldp**

- Use the **no** version to globally remove the LDP configuration.

mpls ldp advertise-labels

- Use to control LDP label distribution.
- You can issue the command one or more times to construct a filter that determines whether and where incoming (locally assigned) labels are distributed. If you do not specify a *toAccessList*, the action is taken for all peers.
- By default, LDP advertises label mappings for all IGP prefixes to all LDP peers. In this case, mappings are not advertised for interface addresses. You can alternatively specify that LDP labels be distributed for a particular interface itself, in addition to the subnet that the interface is on. This behavior enables LSPs to be set up to the LSR configured with the interface address.
- When the LSR learns an IGP route and tries to decide whether to advertise a label for the destination to a particular LDP neighbor, it attempts to match the destination against a route access list specified by this command, in the order in which the commands were issued. The first match determines the action taken, and no further matching is attempted for that destination. If the destination matches, labels are advertised to peers subject to any specified neighbor address list. If either access list is not matched, the labels are not advertised.
- Example

  host1(config)#**mpls ldp advertise-labels for net25 to euro3**
  host1(config)#**mpls ldp advertise-labels for boston1**

  In this example, suppose the LSR receives a label for destination 10.10.11.12. If net25 specifies 10.10.11.12, then the access list action—permit or deny—is taken with the destination. If the action is permit, the peer that the label is advertised to is subject to the access list euro3. If net25 does not include 10.10.11.12, the LSR attempts to match it against boston1. If 10.10.11.12 is present in that access list, the specified action is taken for all peers. If boston1 does not include the destination, the label is not advertised to any peer.

- Use the **no** version to negate label distribution rules according to the specified keywords.

mpls ldp discovery transport-address

- Use to specify the transport address advertised in LDP discovery hello messages for interfaces that use the platform label space. The peer router uses the transport address to establish the session TCP connection.
- Example

  host1(config)#**mpls ldp discovery transport-address 10.21.5.6**

- Use the **no** version to return to using the default transport address, the LSR router ID.

mpls ldp egress-label

- Use to advertise the explicit null label (label 0) or a non-null label, for the LSR that is the egress router for the prefix.
- By default, LDP on the egress router advertises the implicit null label (label 3) for directly connected prefixes. This label causes the router's upstream neighbor to perform a penultimate hop pop.
- Issuing this command automatically creates LDP or MPLS on the current virtual router if they do not yet exist there.
- This command takes effect immediately.
- Example

  host1(config)#**mpls ldp egress-label explicit-null**

- Use the **no** version to restore the default condition, where LDP advertises the implicit null label (label 3) for directly connected prefixes.

mpls ldp independent-control

- Use to specify independent control as the method of label distribution to be used.
- By default, ordered control is used as the method of label distribution for LDP.
- Issuing this command automatically creates LDP or MPLS on the current virtual router if they do not yet exist there.
- This command takes effect immediately.
- Example

  host1(config)#**mpls ldp independent-control**

- Use the **no** version to restore the default condition, wherein ordered control is the label distribution control mode.

mpls ldp ip-forwarding

- Use to specify LSPs to be placed into the IP routing table for forwarding plain IP traffic.
- This command replaces the deprecated **mpls topology-driven-lsp ip-interfaces** command.
- You can specify access lists or prefix lists that describe the LSPs, or include all LSPs in the routing table.
- Example

  host1(config)#**mpls ldp ip-forwarding access-list someLSPs**

- Use the **no** version to prevent the inclusion of the listed LSPs or all LSPs in the routing table.

mpls ldp profile

- Use to create or modify a configuration profile for LDP. Places the CLI in LDP Configuration mode.
- When you issue this command from Global Configuration mode, the **interface** keyword is currently optional in the affirmative version of the command, but mandatory in the **no** version. In a future release it may become mandatory in both versions. We recommend that you use this keyword for both versions.
- If you do not specify a profile name, the factory default profile is assumed. If you specify a profile not previously configured, it is created with the factory default settings.
- Any change to a profile affects all interfaces that use the profile; changes are effective the next time the interface comes up.
- The following values are defined for the implicit default LDP profile:
  - hello hold-time—0 (15 seconds for link hellos; 45 seconds for targeted hello messages)
- Example

  host1(config)#**mpls ldp interface profile ldp1**

- Use the **no** version to delete the profile.

mpls ldp redistribute

- Use to redistribute routes from a specified IGP to LDP, enabling LDP to advertise labeled BGP routes for the inter-AS option C two-stack scenario.
- You can specify a route map to redistribute more specific routes to LDP. Only routes that pass the maps clauses are redistributed. If you do not use a route map for fine-grained control of route distribution, and you do not want to redistribute BGP routes, then you do not need to issue this command.
- For backward compatibility, if you do not issue this command, LDP continues to advertise labels for IGP routes—including connected, static, IS-IS, OSPF, and RIP routes, but excluding BGP routes. That is, IGP routes other than BGP are redistributed to LDP by default.
- Example

  host1(config)#**mpls ldp redistribute ospf route-map boston5**

- Use the **no** version to halt redistribution from the specified IGP or according to the specified route map.

mpls ldp session holdtime

- Use to specify the period for which an LSR maintains the session with its LDP peer without receipt of any LDP message from that peer. The LDP session is terminated when the timer expires.
- Each LSR peer sends the session hold time in its initialization message; peers negotiate to use the minimum of the session hold times proposed by the pair of LSRs. This negotiated session hold time is used by the keepalive timer to maintain the session.

- The keepalive timer is reset with the receipt of any session message from the peer. In the absence of other LDP protocol messages are being sent, peers periodically send a keepalive message to maintain the LDP session.
- Specify a number in the range 15–65535, corresponding to the period in seconds.
- Example

  host1(config)#**mpls ldp session holdtime 1800**

- Use the **no** version to restore the default value, 180 seconds.

mpls ldp session keepalive interval

- Use to specify the interval at which LDP sends session keepalive messages to maintain the LDP session.
- Keepalive messages are sent to LDP peers at this interval in the absence of any other LDP traffic over the session.
- Issuing this command automatically creates LDP or MPLS on the current virtual router if they do not yet exist there.
- This command takes effect immediately.
- Specify a number in the range 1–65535, corresponding to the period in seconds.
- Example

  host1(config)#**mpls ldp session keepalive interval 180**

- Use the **no** version to restore the default value, 20 seconds.

mpls ldp session retries

- Use to specify the number of attempts to be made to set up an MPLS LDP session.
- Specify a number in the range 0–65535. The default value of 0 means the attempts will be made until successful.
- Example

  host1(config)#**mpls ldp session retries 1800**

- Use the **no** version to restore the default value, 0.

mpls ldp session retry-time

- Use to specify the interval in seconds between attempts to set up an MPLS LDP session.
- Specify a number in the range 0–60.
- Example

  host1(config)#**mpls ldp session retry-time 2**

- Use the **no** version to restore the default value, 30 seconds.

mpls ldp targeted-hello holdtime

- Use to specify the period for which a sending LSR maintains a record of targeted hello messages from the receiving LSR without receipt of another targeted hello message from that LSR.
- Each LSR peer sends the hold time in its targeted hello messages; peers negotiate to use the minimum of the hold times proposed by the pair of LSRs.
- The hold timer is restarted whenever the LSR receives a targeted hello from the peer in the targeted hello adjacency. The timer expires if no targeted hello is received from the peer within the hold time. The LSR deletes the targeted hello adjacency when the timer expires. If all targeted hello adjacencies are deleted for an LDP session, then the LSR terminates the LDP session.
- Issuing this command automatically creates LDP or MPLS on the current virtual router if they do not yet exist there.
- This command takes effect immediately.
- Specify a number in the range 1–65535, corresponding to the period in seconds.
- Example

  host1(config)#**mpls ldp targeted-hello holdtime 90**

- Use the **no** version to restore the default value, 45 seconds.

mpls ldp targeted-hello interval

- Use to specify the interval between targeted hello packets sent by LDP.
- Issuing this command automatically creates LDP or MPLS on the current virtual router if they do not yet exist there.
- This command takes effect immediately.
- Specify a number in the range 1–65535, corresponding to the interval in seconds.
- Example

  host1(config)#**mpls ldp targeted-hello interval 30**

- Use the **no** version to restore the default value, 15 seconds.

mpls ldp targeted-hello receive list

- Use to configure the list of peer addresses from which MPLS accepts targeted hello messages.
- This command is unnecessary if you configure the **mpls ldp targeted-hello send list** command.
- You can specify one or more access lists or IP addresses as members of the list.
- If a receive list is configured, hello messages are accepted only from list members. If no list is configured, hello messages are accepted from all addresses.

- Example

  host1(config)#**mpls ldp targeted-hello receive list concord3**

- Use the **no** version to remove the list of peer addresses.

mpls ldp targeted-hello send list

- Use to configure the list of peer addresses to which MPLS sends and accepts targeted hello messages.
- You can specify one or more access lists or IP addresses as members of the list.
- Example

  host1(config)#**mpls ldp targeted-hello send list 10.56.84.21**

- Use the **no** version to remove the list of peer addresses.

mpls topology-driven-lsp

- Use to turn on topology-driven LSP, where the LSR automatically creates LSPs when it learns a new IGP route.
- The router advertises labels for new routes immediately to all peers without waiting for a label request message from an upstream peer or a label mapping message from a downstream peer. This mode is downstream-unsolicited, independent control.
- Example

  host1(config)#**mpls topology-driven-lsp**

- Use the **no** version to disable topology-driven LSPs.

RSVP-TE Tasks and Commands

Typically, you do not configure anything for RSVP-TE at the global level, but you can perform the following optional tasks.

1. (Optional) Enable RSVP-TE. Any RSVP-TE–related command creates RSVP-TE implicitly, negating the need to issue this command.

   host1(config)#**mpls rsvp**

2. (Optional) Configure retry timer options globally (to apply to all tunnels) to set up an LSP after a setup failure. You can also configure timers for a specific tunnel; see *Tunnel Configuration Tasks* on page 250.

   host1(config)#**mpls lsp retries 35**
   host1(config)#**mpls lsp retry-time 55**

3. (Optional) Configure the interval at which the bandwidth values are flooded.

   host1(config)#**mpls traffic-eng link-management timers periodic-flooding 10**

4. (Optional) Configure reoptimization—How often MPLS searches for better paths for the tunnel.

   host1(config)#**mpls reoptimize timers frequency 180**

   You can also force an immediate search for better paths for all existing LSPs.

   host1#**mpls reoptimize**

5. (Optional) Enable refresh reduction and message bundling.

   host1(config)#**mpls rsvp refresh-reduction**
   host1(config)#**mpls rsvp message-bundling**

6. (Optional) Configure the egress router to advertise the explicit null label.

   host1(config)#**mpls rsvp egress-label explicit-null**

mpls lsp no-route retries

- Use to specify the number of attempts to be made to set up an RSVP-TE tunnel after a failure due to no available route.
- Specify a number in the range 0–65535. The default value of 0 means the attempts will be made until successful.
- Example

  host1(config)#**mpls lsp no-route retries 3200**

- Use the **no** version to restore the default value, 0.

mpls lsp no-route retry-time

- Use to specify the interval in seconds between attempts to set up an RSVP-TE tunnel after a failure due to no available route.
- Specify a number in the range 1–60.
- Example

  host1(config)#**mpls lsp no-route retry-time 45**

- Use the **no** version to restore the default value, 5 seconds.

mpls lsp retries

- Use to specify the number of attempts to be made to set up an RSVP-TE tunnel after a failure *other* than one due to no available route.
- Specify a number in the range 0–65535. The default value of 0 means the attempts will be made until successful.
- Example

  host1(config)#**mpls lsp retries 40**

- Use the **no** version to restore the default value, 0.

mpls lsp retry-time

- Use to specify the interval in seconds between attempts to set up an RSVP-TE tunnel after a failure *other* than one due to no available route.
- Specify a number in the range 1–60.
- Example

  host1(config)#**mpls lsp retry-time 15**

- Use the **no** version to restore the default value, 5 seconds.

mpls reoptimize

- Use to perform an immediate search for better paths for all existing tunnels.
- Example

  host1#**mpls reoptimize**

- There is no **no** version.

mpls reoptimize timers frequency

- Use to specify the frequency with which existing tunnels are searched for better paths.
- Specify a number in the range 0–604800, corresponding to the period between searches in seconds.
- A value of 0 means no reoptimization is performed.
- Example

  host1(config)#**mpls reoptimize timers frequency 86400**

- Use the **no** version to restore the default value, 3600 seconds.

mpls rsvp

- Use to enable RSVP-TE.
- Because any RSVP-TE–related command creates RSVP-TE implicitly, you do not need to issue this command.
- Example

  host1(config)#**mpls rsvp**

- Use the **no** version to disable RSVP-TE.

mpls rsvp egress-label

- Use to configure the egress router to advertise the explicit null label or a real label. This advertisement ensures that packets received from upstream include a label and that the egress router performs ultimate hop popping.
- Use the **explicit-null** keyword to advertise the explicit null label. Use the **non-null** label to advertise a real label.
- Example

  host1(config)#**mpls rsvp egress-label explicit-null**

- Use the **no** version to restore the default value, where the egress router advertises the implicit null label.

mpls rsvp message-bundling

- Use to enable RSVP-TE to send bundle messages, each of which includes multiple standard RSVP-TE messages, to reduce the overall message processing overhead.
- Example

  host1(config)#**mpls rsvp message-bundling**

- Use the **no** version to disable RSVP-TE message bundling.

mpls rsvp profile

- Use to create or modify a configuration profile for RSVP-TE. Places the CLI in RSVP Configuration mode.
- When you issue this command from Global Configuration mode, the **interface** keyword is currently optional in the affirmative version of the command, but mandatory in the **no** version. In a future release it may become mandatory in both versions. We recommend that you use this keyword for both versions.
- If you do not specify a profile name, the factory default profile is assumed. If you specify a profile not previously configured, it is created with the factory default settings.
- Any change to a profile affects all interfaces that use the profile; changes are effected the next time the interface comes up.
- The following values are defined for the implicit default RSVP profile:
  - refresh period—30,000 ms (30 seconds)
  - timeout factor—3
- Example

  host1(config)#**mpls rsvp interface profile rsvp1**

- Use the **no** version to delete the profile.

mpls rsvp refresh-reduction

- Use to enable RSVP-TE summary refresh and reliability features, including the message ID object, the message ack object, and summary refresh messages.
- Example

  host1(config)#**mpls rsvp refresh-reduction**

- Use the **no** version to disable summary refresh and reliability.

mpls traffic-eng link-management timers periodic-flooding

- Use to specify in seconds how often the bandwidth change is flooded throughout the network.
- To turn periodic flooding off, set the value to 0.
- Specify a number in the range 0–3600.
- Example

  host1(config)#**mpls traffic-eng link-management timers periodic-flooding 240**

- Use the **no** version to restore the default value, 180 seconds.

BGP Tasks

For BGP global configuration tasks, see *Chapter 1, Configuring BGP Routing* and *Chapter 3, Configuring BGP-MPLS Applications*.

LDP Interface Profile Configuration Tasks and Command

Complete these optional tasks to configure the label distribution options. Creating or accessing an interface profile places the CLI in LDP Configuration mode. When you have completed the interface profile configuration tasks, proceed to the section *Interface Configuration Tasks* on page 244.

1. Access the desired profile configuration mode.

   host1(config)#**mpls ldp interface profile ldp5**

2. Configure interface profile settings, changing the values from the implicit default values.

   host1(config-ldp)#**hello hold-time 30**
   host1(config-ldp)#**hello interval 10**

hello hold-time

- Use to specify the period for which a sending LSR maintains a record of link hello messages from the receiving LSR without receipt of another link hello message from that LSR.
- Each LSR peer sends the hold time in its link hello messages; peers negotiate to use the minimum of the hold times proposed by all LSRs on the same subnet.

- The hold timer is restarted whenever the LSR receives a link hello from the adjacent peer. The timer expires if no link hello is received from the adjacent peer within the hold time. The LSR deletes the link hello adjacency when the timer expires. If all link hello adjacencies are deleted for an LDP session, then the LSR terminates the LDP session.
- Specify a number in the range 1–65535, corresponding to the period in seconds.
- Example

  host1(config-ldp)#**hello hold-time 55**

- Use the **no** version to restore the default value, 15 seconds.

hello interval

- Use to specify the interval between link hello packets sent by LDP.
- Specify a number in the range 1–65535, corresponding to the interval in seconds.
- Example

  host1(config-ldp)#**hello interval 10**

- Use the **no** version to restore the default interval, 5 seconds.

RSVP Interface Profile Configuration Tasks and Commands

Complete these optional tasks to configure the label distribution options. Creating or accessing an interface profile places the CLI in RSVP Configuration mode. When you have completed the interface profile configuration tasks, proceed to the section *Interface Configuration Tasks* on page 244.

1. Access the desired profile configuration mode.

   host1(config)#**mpls rsvp interface profile rsvp4**

2. Configure interface profile settings, changing the values from the implicit default values.

   host1(config-rsvp)#**refresh-period**
   host1(config-rsvp)#**cleanup-timeout-factor**

cleanup-timeout-factor

- Use to specify the number of refresh messages that can be lost before the path or resv state is ended.
- Together with the refresh period, defines the RSVP tunnel timeout period. See *RSVP-TE Messages and Sessions* on page 219 for more information.
- Example

  host1(config-rsvp)#**cleanup-timeout-factor 9**

- Use the **no** version to restore the default value of 3.

refresh-period

- Use to specify the timeout period in milliseconds between generation of refresh messages.
- Specify a value in the range 0–4294967295; the default value is 30,000 milliseconds.
- Together with the cleanup timeout factor, defines the RSVP tunnel timeout period. See *RSVP-TE Messages and Sessions* on page 219 for more information.
- Example

  host1(config-rsvp)#**refresh-period 60000**

- Use the **no** version to restore the default value, 30000 milliseconds.

Interface Configuration Tasks

These tasks are performed at the major interface over which you want to run MPLS. Creating or accessing an interface places the CLI in Interface Configuration mode. You can then configure MPLS options on that interface. The following sequence is arbitrary; you can perform these tasks in any order.

NOTE: Loop detection is always enabled in the JUNOSe MPLS implementation.

When you have completed the interface configuration tasks, proceed to *Tunnel Configuration Tasks* on page 250.

MPLS Tasks and Commands

1. Enable MPLS on the interface.

   host1(config-if)#**mpls**

   or

   host1(config-if)#**no mpls disable**

2. (Optional) Configure the interface label space with the VPI and VCI ranges.

   host1(config-if)#**mpls atm vpi range 10 200**
   host1(config-if)#**mpls atm vci range 33 4000**

   Only ATM AAL5 interfaces support the interface label space.

3. (Optional) Specify an interface for signaling for an MPLS major interface in the interface label space.

   host1(config-if)#**mpls signaling-interface atm 4/0.5**

mpls

- Use to create or delete MPLS on an interface.
- MPLS interfaces are also implicitly created by any other MPLS commands issued in Interface Configuration mode.
- Example

  host1(config-if)#**mpls**

- Use the **no** version to halt MPLS on the interface and delete the MPLS interface configuration.

mpls atm vci range

- Use to specify the range for virtual circuit identifiers for LSPs on ATM AAL5 major interfaces.
- An interface can support RSVP-TE as the label distribution protocol For MPLS interfaces using the interface label space, you must configure the both the VPI and VCI range. If you do not, the interface will remain operationally down.
- Specify the lowest and highest allowed VCI numbers, each in the range 33–65535.
- Example

  host1(config-if)**#mpls atm vci range 45 4000**

- Use the **no** version to remove the range.

mpls atm vpi range

- Use to specify the range for virtual path identifiers for LSPs on ATM AAL5 major interfaces.
- An interface can support RSVP-TE as the label distribution protocol For MPLS interfaces using the interface label space, you must configure the both the VPI and VCI range. If you do not, the interface will remain operationally down.
- Specify the lowest and highest allowed VPI numbers, each in the range 0–255.
- Example

  host1(config-if)#**mpls atm vpi range 10 200**

- Use the **no** version to remove the range.

mpls disable

- Use to bring the interface administratively down.
- Example

  host1(config-if)#**mpls disable**

- Use the **no** version to bring the interface administratively up.

mpls signaling-interface

- Use to specify a layer 2 interface for an MPLS major interface. MPLS uses the IPv4 interfaces stacked on that layer 2 interface as the signaling interface for this MPLS major interface.
- Typically, you issue this command for MPLS major interfaces stacked on ATM AAL5 interfaces. For example, multiple ATM1483 interfaces might be stacked on the ATM AAL5 interface. You can issue this command to select one of these ATM1483 interfaces for signaling.
- If you do not issue this command, then MPLS nondeterministically selects a layer 2 interface for signaling.
- You can configure this command only for interfaces that use the interface label space. For interfaces that use the platform label space, the signaling interface is always next to the MPLS major interface, that is, stacked on the same lower interface and is not configurable.
- Example

  host1(config-if)#**mpls signaling-interface atm 4/0.5**

- Use the **no** version to restore the default behavior, wherein MPLS nondeterministically selects a layer 2 interface for signaling.

LDP Tasks and Command

1. Start LDP on the interface.
   - Using the default values (an implicit *default* profile):

     host1(config-if)#**mpls ldp**

   - Using a previously created profile:

     host1(config-if)#**mpls ldp profile ldp5**

2. (Optional) Suppress transmission of link hello messages to all LSRs.

   host1(config-if)#**mpls ldp link-hello disable**

mpls ldp disable

- Use to disable LDP on the interface.
- Example

  host1(config-if)#**mpls ldp disable**

- Use the **no** version to reenable LDP on the interface.

mpls ldp link-hello disable

- Use to suppress the transmission of LDP link hello messages.
- This command requires the use of targeted hello messages to form LDP peer adjacencies.
- Example

  host1(config-if)#**mpls ldp link-hello disable**

- Use the **no** version to restore the default, where the LSR sends multicast link hello messages.

mpls ldp profile

- Use to configure LDP on the interface.
- You can specify a profile name. If you specify a profile not previously configured, it is created with the factory default settings. If you do not specify a profile, the factory-supplied default profile values are used.
- Example

  host1(config-if)#**mpls ldp profile ldp45**

- Use the **no** version to revert to the default profile on the interface.

RSVP-TE Tasks and Commands

1. Start RSVP-TE on the interface.

   - Using the default values (an implicit *default* profile):

     host1(config-if)#**mpls rsvp**

   - Using a previously created profile:

     host1(config-if)#**mpls rsvp profile rsvp4**

2. (Optional) Configure total bandwidth available on the interface.

   host1(config-if)#**bandwidth 8192**

3. (Optional) Configure total bandwidth reservable for MPLS on the interface.

   host1(config-if)#**mpls bandwidth 4096**

4. (Optional) Specify thresholds that trigger bandwidth flooding when crossed by an increase or decrease in the total reservable bandwidth.

   host1(config-if)#**mpls traffic-eng flood thresholds up 15**
   host1(config-if)#**mpls traffic-eng flood thresholds down 15**

5. (Optional) Specify the resource attributes for the interface so that tunnels can discriminate among interfaces.

   host1(config-if)#**mpls traffic-eng attribute-flags 0x000001f9**

6. (Optional) Configure an administrative weight for the interface that overrides the weight assigned by the IGP.

   host1(config-if)#**mpls traffic-eng administrative-weight 25**

bandwidth

- Use to specify the total bandwidth in kilobits per second available on the interface.
- Example

  host1(config-if)#**bandwidth 262144**

- Use the **no** version to remove the admission control configuration (all internal CAC records) from the interface.

mpls bandwidth

- Use to specify the total bandwidth in kilobits per second *reservable* for MPLS on the interface.
- Use the **mpls** version of the command for the JUNOSe implementation.
- Use the **ip rsvp** version of the command for compatibility with implementations on routers from other vendors.
- Example

  host1(config-if)#**mpls bandwidth 32768**

- Use the **no** version to restore the default value, 0.

mpls rsvp disable

- Use to disable RSVP-TE on the interface.
- Example

  host1(config-if)#**mpls rsvp disable**

- Use the **no** version to reenable RSVP-TE on the interface.

mpls rsvp profile

- Use to configure RSVP-TE on the interface.
- You can specify a profile name. If you specify a profile not previously configured, it is created with the factory default settings. If you do not specify a profile, the factory-supplied default profile values are used.
- Example

  host1(config-if)#**mpls rsvp profile ldp45**

- Use the **no** version to revert to the default profile on the interface.

mpls traffic-eng administrative-weight

- Use to specify an administrative weight for the interface.
- For MPLS traffic-engineering purposes, this value supersedes any weight conferred upon the link by the IGP and can be in the range 0–4294967295.
- Example

  host1(config-if)#**mpls traffic-eng administrative-weight 150**

- Use the **no** version to restore the default value, which matches the IGP-determined weight.

mpls traffic-eng attribute-flags

- Use to specify traffic-engineering attributes for an interface; attributes are compared with tunnel affinity bits to determine links eligibility for the tunnel.
- Specify a hexadecimal value in the range 0x0–0xFFFFFFFF. The attribute values have only the meaning that you assign to them; they serve to place the interface within one or more resource classes.
- Example

  host1(config-if)#**mpls traffic-eng attribute-flags 0x100F0010**

- Use the **no** version to restore the default value, 0x0.

mpls traffic-eng flood thresholds

- Use to specify the thresholds that trigger the flooding of the current reservable bandwidth throughout the network.
- Bandwidth is flooded when the percentage of total reservable bandwidth increases past any up threshold or decreases past any down threshold.
- You can list more than one percentage; flooding is triggered by all percentages specified.
- Example

  host1(config-if)#**mpls traffic-eng flood thresholds up 25**
  host1(config-if)#**mpls traffic-eng flood thresholds down 25**

- Use the **no** version to restore the default values:
  - For increases in bandwidth—15, 30, 45, 60, 75, 80, 85, 90, 95, 97, 98, 99, 100
  - For decreases in bandwidth—100, 99, 98, 97, 96, 95, 90, 85, 80, 75, 60, 45, 30, 15

Tunnel Configuration Tasks

Complete the following tasks to configure a tunnel interface. Configure the tunnel endpoint last; anything configured after the tunnel endpoint does not take effect until the tunnel is brought up the next time. You can perform all other tasks in any order.

NOTE: Tunnel configuration tasks are relevant only for traffic engineering networks.

1. Create the MPLS tunnel interface.

   host1(config)#**interface tunnel mpls:boston**

2. (Optional) Configure the LSP to announce its endpoint to an IGP (sometimes referred to as *registering the endpoint*).

   host1(config-if)#**tunnel mpls autoroute announce isis**

3. (Optional) Specify a tunnel metric to be used by an IGP in its SPF calculation.

   host1(config-if)#**tunnel mpls autoroute metric absolute 100**

4. (Optional) Configure the path options used for the tunnel.

   host1(config-if)#**tunnel mpls path-option 3 dynamic isis**

5. (Optional) Configure the bandwidth required for the tunnel.

   host1(config-if)#**tunnel mpls bandwidth 1240**

6. (Optional) Configure preemption hold or setup priority.

   host1(config-if)#**tunnel mpls traffic-eng priority 4 4**

7. (Optional) Configure resource class affinity.

   host1(config-if)#**tunnel mpls traffic-eng affinity 0x1100 mask 0xFFFF**

8. (Optional) Configure retry timers options to apply to a specific tunnel to set up an LSP after a setup failure.

   host1(config-if)#**tunnel mpls no-route retries 100**
   host1(config-if)#**tunnel mpls no-route retry-time 45**
   host1(config-if)#**tunnel mpls retries 250**
   host1(config-if)#**tunnel mpls retry-time 65**

9. (Optional) Associate a text description with the tunnel.

   host1(config-if)#**tunnel mpls description southshore**

10. Configure the tunnel endpoint.

    host1(config-if)#**tunnel destination 10.12.21.5**

interface tunnel

- Use to create a tunnel interface for MPLS.
- You can specify that the tunnel be established in the routing space of a virtual router other than the current VR. If you specify another VR, all MPLS tunnel commands apply to the tunnel in that VR. If you do not specify another VR, tunnel commands apply to the current VR.
- Example 1—tunnel in current VR

  host1(config)#**interface tunnel mpls:5**

- Example 2—tunnel in another VR

  host1(config)#**interface tunnel mpls:5 transport-virtual-router vr5**

- Use the **no** version to remove the tunnel interface.

tunnel destination

- Use to configure the tunnel endpoint for the tunnel.
- The IP address for the destination must be one of the following if the destination is on another E-series router:
  - The IP address of the interface that has MPLS enabled
  - The router ID of the destination router
- Example

  host1(config-if)#**tunnel destination 10.12.21.**

- Use the **no** version to delete the endpoint.

tunnel mpls affinity

- Use to configure an affinity constraint on a given tunnel.
- The affinity value specifies the class of resources—resource attributes—associated with the tunnel. Affinity values range from 0x0 to 0xFFFFFFFF; the default is 0x0.
- Applying the mask to the affinity bits determines the value of the attribute flags that a link (interface) must have in order to be used by a tunnel. If a mask bit is one, the corresponding interface attribute flag must match the tunnel's corresponding affinity bit. If a mask bit is zero, the attribute flag does not have to match the tunnel's affinity bit. Mask values range from 0x0 to 0xFFFFFFFF; the default is 0x0000FFFF.
- In the current release, RSVP-TE does not include the affinity bits in its messages. As a result RSVP-TE cannot use any configured affinity restraint for portions of tunnels beyond the ingress router's local area.
- Example 1—matches only links configured with attribute flags 0x000C300*0*

  host1(config-if)#**tunnel mpls affinity 0x000C3000 0xFFFFFFFF**

- Example 2—matches only links configured with attribute flags 0x000C300*0* or 0x000C300*1*

  host1(config-if)#**tunnel mpls affinity 0x000C3000 0xFFFFFFFE**

- Example 3—matches only links configured with attribute flags from 0x000C3000 to 0x000C3003

  host1(config-if)#**tunnel mpls affinity 0x000C3000 0xFFFFFFFC**

- Example 4—matches only links configured with attribute flags from 0x000C3000 to 0x000C300F

  host1(config-if)#**tunnel mpls affinity 0x000C3000 0xFFFFFFF0**

- Use the **no** version to delete the affinity constraint from the tunnel.

tunnel mpls autoroute announce

- Use to configure the LSP to register its endpoint (the egress router) with the configured routing protocol—IS-IS or OSPF—so that the protocol can use the tunnel to determine routes.
- If you do not specify a routing protocol, the default behavior is to announce to both IS-IS and OSPF.
- Example

  host1(config-if)#**tunnel mpls autoroute announce isis**

- Use the **no** version to disable endpoint announcements.

tunnel mpls autoroute metric

- Use to specify the tunnel metric. The value determines tunnel preference when there is more than one tunnel or native IP path to a tunnel endpoint. A lower value is preferred to a higher value.
- If you set up multiple tunnels, when the primary tunnel goes down, the existing tunnel with the lowest metric is used immediately.
- If you specify an absolute value in the range 1–2147483647, this value overrides the metric for the path provided by the IGP.
- If you specify a relative value from -10 to +10, this value is subtracted from (-) or added to (+) the metric for the path provided by the IGP.
- Example

  host1(config-if)#**tunnel mpls autoroute metric relative -5**

- Use the **no** version to restore the default value, relative 0, meaning that the tunnel metric is the IGP value.

tunnel mpls bandwidth

- Use to configure a bandwidth constraint on a given tunnel in kilobits per second.
- When you specify bandwidth for a traffic engineering tunnel, MPLS automatically creates and attaches a QoS profile to the tunnel to enforce the bandwidth reservation.

  The QoS profile creates a scheduler node at the LSP level, with a scheduler profile that has an assured rate corresponding to the reserved bandwidth.

The QoS profile also creates queues above the scheduler node so that traffic of a particular class will be subject to the scheduler node. If no queue are created at the LSP level for a particular class, the traffic of that class enters that class's queue at a lower level, bypassing the bandwidth reservation enforcement.

- Example

  host1(config-if)#**tunnel mpls bandwidth 2148**

- Use the **no** version to delete the bandwidth constraint from the tunnel.

tunnel mpls description

- Use to associate a text description with the tunnel.
- Specify a string of up to 40 alphanumeric characters.
- Example

  host1(config-if)#**tunnel mpls description boston2dc**

- Use the **no** version to delete the description.

tunnel mpls no-route retries

- Use to specify the number of attempts to be made to set up an RSVP-TE tunnel after a failure due to no available route.
- Specify a number in the range 0–65535. The default value of 0 means the attempts will be made until successful.
- Example

  host1(config-if)#**tunnel mpls no-route retries 3200**

- Use the **no** version to restore the default value, 0.

tunnel mpls no-route retry-time

- Use to specify the interval in seconds between attempts to set up an RSVP-TE tunnel after a failure due to no available route.
- Specify a number in the range 1–60.
- Example

  host1(config-if)#**tunnel mpls no-route retry-time 45**

- Use the **no** version to restore the default value, 5 seconds.

tunnel mpls path-option

- Use to specify the path options for the tunnel. You can configure one or more path options—each identified by a unique number—for a given tunnel.
- Options include whether the path is dynamic or explicit and whether hop-by-hop, IS-IS, or OSPF routing is used.
- Options with a lower number are preferred; path option 1 is the most preferred. If an LSP goes down on the currently used path option, the path option with the next highest preference is used.

- Example

  host1(config-if)#**tunnel mpls path-option 3 dynamic isis**

- Use the **no** version to delete the path options.

tunnel mpls priority

- Use to configure a priority for a given tunnel, ranging from the highest priority of 0 to the lowest priority of 7.
- You can configure a setup priority or both a setup priority and a hold priority for the tunnel:
  - Setup priority—Priority of an LSP while it is being established; default value is 4
  - Hold priority—Priority of an LSP after it has been established; default value is equal to the specified setup priority

  If insufficient resources exist for a new LSP to be established, its setup priority is evaluated against the hold priorities of existing LSPs. If the new LSP has a higher priority, it preempts the resources from the lower-priority existing LSP(s) and is established.
- A setup priority of 0 can preempt all nonzero hold priorities.
- The setup priority cannot be better (lower numerically) than the hold priority. For example, if the setup priority for a tunnel is 2, the hold priority must be 2, 1, or 0.
- Example

  host1(config-if)#**tunnel mpls priority 3 2**

- Use the **no** version to change the priority to the default value, 4 for both setup and hold priorities.

tunnel mpls retries

- Use to specify the number of attempts to be made to set up an RSVP-TE tunnel after a failure *other* than one due to no available route.
- Specify a number in the range 0–65535. The default value of 0 means the attempts will be made until successful.
- Example

  host1(config-if)#**tunnel mpls retries 1275**

- Use the **no** version to restore the default value, 0.

tunnel mpls retry-time

- Use to specify the interval in seconds between attempts to set up an RSVP-TE tunnel after a failure *other* than one due to no available route.
- Specify a number in the range 1–60.
- Example

  host1(config-if)#**tunnel mpls retry-time 15**

- Use the **no** version to restore the default value, 5 seconds.

Tunnel Profile Configuration Tasks

If you anticipate having multiple tunnels to share the same configuration, you can reduce your configuration time by using tunnel profiles to configure your tunnels. When you create a tunnel profile, you can use most of the commands presented in *Tunnel Configuration Tasks* on page 250; see that section for descriptions of the commands.

In the profile, configure the tunnel endpoint last; anything configured after the tunnel endpoint does not take effect until the tunnel is brought up the next time. You can perform all other tasks in any order.

NOTE: Tunnel profile configuration tasks are relevant only for traffic engineering networks.

To configure a tunnel profile:

1. Enter Tunnel Profile Configuration mode.

   host1(config)#**mpls tunnels profile Lisbon**

2. (Optional) Configure the LSP to announce its endpoint to an IGP.

   host1(config-tunnelprofile)#**tunnel mpls autoroute announce isis**

3. (Optional) Specify a tunnel metric to be used by an IGP in its SPF calculation.

   host1(config-tunnelprofile)#**tunnel mpls autoroute metric absolute 100**

4. (Optional) Configure the path options used for the tunnel.

   host1(config-tunnelprofile)#**tunnel mpls path-option 3 dynamic isis**

5. (Optional) Configure the bandwidth required for the tunnel.

   host1(config-tunnelprofile)#**tunnel mpls bandwidth 1240**

6. (Optional) Configure preemption hold or setup priority.

   host1(config-tunnelprofile)#**tunnel mpls priority 4 4**

7. (Optional) Configure resource class affinity.

   host1(config-tunnelprofile)#**tunnel mpls affinity 0x1100 mask 0xFFFF**

8. (Optional) Configure retry timers options to apply to a specific tunnel to set up an LSP after a setup failure.

   host1(config-tunnelprofile)#**tunnel mpls no-route retries 100**
   host1(config-tunnelprofile)#**tunnel mpls no-route retry-time 45**
   host1(config-tunnelprofile)#**tunnel mpls retries 250**
   host1(config-tunnelprofile)#**tunnel mpls retry-time 65**

9. (Optional) Associate a text description with the tunnel.

   host1(config-tunnelprofile)#**tunnel mpls description southshore**

10. Configure the tunnel endpoint.

    - For static tunnels

      host1(config-tunnelprofile)#**tunnel destination 10.1.2.5 10.1.2.6**

      All tunnels to the specified destination(s) are configured with the profile settings.

    - For dynamic tunnels

      host1(config-tunnelprofile)#**tunnel destination isis-level-2 access-list madrid3**

      When an endpoint is dynamically learned from the specified routing protocol, MPLS searches its tunnel profiles for a match. The dynamic tunnel is established using the settings from the first matching profile.

mpls tunnels profile

- Use to create a tunnel profile for MPLS.
- A tunnel profile is used for all tunnels sharing the same configuration.
- Example

  host1(config)#**mpls tunnels profile Paris**

- Use the **no mpls tunnels profile** version to delete the tunnel profile. Use the **no mpls tunnels profile disable** version to reenable tunnel profiles previously disabled. When a tunnel profile is disabled, all its associated tunnels are disabled.

tunnel destination

- Use to configure the tunnel endpoints (destinations) associated with the profile.
- If you specify that the endpoints are to be learned from IS-IS or OSPF, tunnels are created when the destinations are learned from the specified IGP. you can filter learned addresses by specifying an access list or a prefix list.
- If you specify a sequence of one or more individual IP addresses as endpoints, tunnels are created as soon as the destination addresses are configured.

- If you specify one or more IP addresses for the destination(s), each address must be one of the following if it is on another E-series router:
  - The IP address of the interface that has MPLS enabled
  - The router ID of the destination router
- Examples

  host1(config-tunnelprofile)#**tunnel destination isis-level-2 prefix-list tunnel5**
  host1(config-tunnelprofile)#**tunnel destination 10.12.25.1 10.12.25.2**

- Use the **no** version to delete the endpoints.

Explicit Routing

MPLS offers two options for selecting routing paths:

- Hop-by-hop routing
- Explicit routing

In explicit routing, the route the LSP takes is defined by the ingress node. The path consists of a series of hops defined by the ingress LSR. Each hop can be a traditional interface, an autonomous system, or an LSP. A hop can be *strict* or *loose*.

A *strict* hop must be directly connected (that is, adjacent) to the previous node in the path. A *loose* hop is not necessarily directly connected to the previous node; whether it is directly connected is unknown.

The sequence of hops comprising an explicit routing LSP may be chosen in either of the following ways:

- Through a user-defined configuration, resulting in *configured* explicit paths. When you create the explicit route, you must manually configure each hop in the path.
- Through a routing protocol–defined configuration, resulting in *dynamic* explicit paths. When the routing protocol (IS-IS or OSPF) creates the explicit path, it makes use of the topological information learned from a link-state database in order to compute the entire path, beginning at the ingress node and ending at the egress node.

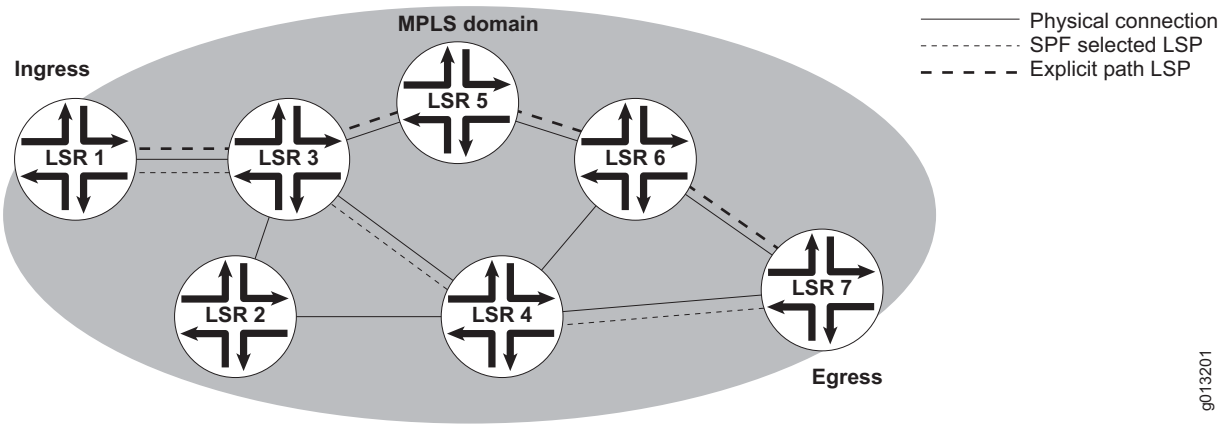
Consider the MPLS domain shown in Figure 58. Without explicit path routing, the tunnel is created hop by hop along the following path:

LSR 1 –> LSR 3 –> LSR 4 –> LSR 7

Suppose LSR 5 and LSR 6 are underused and LSR 4 is overused. In this case you might choose to configure the following explicit path because it forwards the data better than the hop-by-hop path:

LSR 1 –> LSR 3 –> LSR 5 –> LSR 6 –> LSR 7

Figure 58: Explicit Routing in an MPLS Domain



Defining Configured Explicit Paths

You can create explicit routing paths *manually* by configuring an explicit path with a name and a series of addresses (hops) from ingress to egress.

To manually configure explicit routing:

1. Access Explicit Path Configuration mode.

   host1(config)#**mpls explicit-path name xyz**
   host1(config-expl-path)#

2. Do one of the following to configure the hops in the LSP:

   - Set the next hop (if need be) at a particular index in the explicit path.
   - Add the next hop (if need be) after a particular index in the explicit path.

3. Enable the explicit path.

NOTE: To prevent a partially configured explicit path from being used, do not enable it until you have finished configuring or modifying the path.

4. (Optional) List the currently configured explicit path.

append-after

- Use to add a next hop after a particular index in the explicit path.
- Sequence (index) numbers after this index adjust automatically.
- Example

  host1(config-expl-path)#**append-after 5 next-address 192.168.47.22**

- There is no **no** version.

index

- Use to set a next hop at a particular index in the explicit path.
- Example

  host1(config-expl-path)#**index 5 next-address 172.18.100.5**

- Use the **no** version to remove the next hop from the index.

list

- Use to list the currently configured explicit path (optionally starting at a particular index defined with the **index** command).
- Example

  host1(config-expl-path)#**list 5**

- There is no **no** version.

mpls explicit-path

- Use to define an explicit path by name and also to enable or disable an explicit path.
- Also accepts the **ip** keyword instead of **mpls** for compatibility with non–E-series implementations.
- Examples

  host1(config)#**mpls explicit-path name xyz**
  host1(config)#**ip explicit-path name xyz enable**

- Use the **no** version to delete the specified explicit path.

next-address

- Use to configure an IPv4 hop at the end of the explicit path.
- You can specify a loose node, which indicates that the node is not necessarily directly connected (adjacent) to the previous node in the path. If you do not specify loose, the configuration defaults to strict, indicating that the node is directly connected to the previous node.
- Example

  host1(config-expl-path)#**next-address 10.10.9.2**

- There is no **no** version.

Specifying Configured Explicit Paths on a Tunnel

After you have defined a configured explicit path, you can configure the path on a tunnel.

To configure explicit routing on a tunnel:

1. Create an MPLS tunnel.

   host1(config)#**interface tunnel mpls:1**

2. Set the path option.

   host1(config-if)#**tunnel mpls path-option 1 explicit name xyz**

tunnel mpls path-option

- Use to specify the path options for the tunnel. You can configure one or more sets of path options—each identified by a unique number—for a given tunnel.
- Options with a lower number are preferred; path option 1 is the most preferred. If an LSP goes down on the currently used path option, the path option with the next highest preference is used.
- Example 1—configured tunnel:

  host1(config-if)#**tunnel mpls path-option 1 explicit name xyz**

- Example 2—dynamic tunnel:

  host1(config-if)#**tunnel mpls path-option 2 dynamic isis**

- Use the **no** version to delete the path options.

Configuring Dynamic Explicit Paths on a Tunnel

You can create explicit routing paths *dynamically* with a routing protocol. IS-IS and OSPF both currently support explicit routing.

To configure dynamic explicit routing:

1. Create an MPLS tunnel.

   host1(config)#**interface tunnel mpls:bilbao5**

2. Set the path option.

   host1(config-if)#**tunnel mpls path-option 2 dynamic isis**

To configure MPLS dynamic explicit path routing, refer to the commands and guidelines in the previous section, *Specifying Configured Explicit Paths on a Tunnel*.

Monitoring Explicit Paths

For information about monitoring MPLS explicit routing, see the following sections in this chapter:

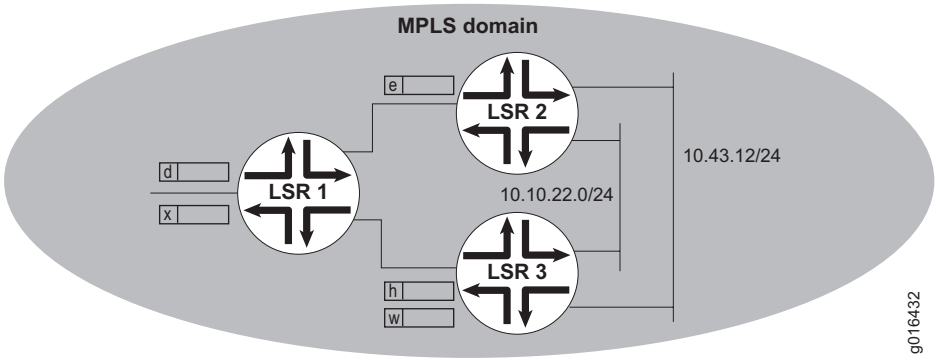
- *Monitoring MPLS* on page 318
- *Configuring IGPs and MPLS* on page 298

Configuring LDP FEC Deaggregation

Beginning with JUNOSe Release 8.1.0, LDP routers running JUNOSe employ LDP FEC aggregation by default. FEC aggregation means that when an LDP egress router advertises multiple prefixes, all the prefixes are members of the same FEC. Only a single label is advertised for this FEC. LDP maintains this aggregation as the advertisement traverses the network, if possible.

Consider the topology shown in Figure 59.

Figure 59: FEC Aggregation and Equal-Cost Paths



In this example, LSR 2 uses FEC aggregation, but LSR 3 does not. Consequently, LSR 2 advertises the single label *e*, mapped to a FEC that includes both prefixes, 10.10.22.0/24 and 10.43.12.0/24.

In contrast, LSR 3 has two FECs, one for 10.10.22.0/24 and one for 10.43.12.0/24. A separate label is bound to each FEC. LSR 3 advertises label *h* for one FEC and label *w* for the other FEC.

LSR 2 and LSR 3 are downstream routers for LSR 1. LSR 1 does not aggregate. Instead, LSR 1 advertises label *d* for 10.10.22.0/24 and label *x* for 10.43.12.0/24.

mpls ldp deaggregate

- Use to bind each prefix on the current virtual router to a separate label.
- Earlier versions of the JUNOSe software do not support FEC aggregation. When interoperating with routers running such a release, you must issue this command
- Issuing this command automatically creates LDP or MPLS on the current virtual router if they do not yet exist there.
- This command takes effect immediately.

- Example

  host1(config)#**mpls ldp deaggregate**

- Use the **no** version to restore the default condition, where LDP aggregates multiple prefixes to be bound to the same label.

Configuring LDP Graceful Restart

The graceful restart mechanism minimizes the negative effect on MPLS forwarding across an LSR restart. You can configure the neighbors of the LSR to wait for the LSR to restart (helper mode). When the LSR restarts, the neighbors mark their current label mapping entries from the LSR as stale. If the LSR recovers within the proper interval, the entries are no longer marked as stale and are used as they were before the LDP connection failed. If the LSR does not recover in time, the entries are deleted.

LDP graceful restart supports only the downstream-unsolicited mode of label distribution. Successful operation of LDP graceful restart requires that stateful SRP switchover (high availability) be configured on the router. Although you can configure LDP graceful restart if stateful SRP switchover is not configured on the router, the graceful restart capability will not function.

You can configure an LSR to restart itself gracefully and to support graceful restart in its neighbors (helper mode), or helper mode alone. In either case, the LSR includes the fault tolerant (FT) session TLV in the LDP initialization messages it sends at session startup. The TLV includes values for the reconnect timeout and the recovery time. When both graceful restart and the helper mode are disabled, the LSR does not include the TLV in its LDP initialization messages.

The configurable reconnect time specifies how long you want the LSR's neighbors to wait for the LSR to resume exchanging LDP messages with the neighbors after the connection failure. The reconnect timeout value is nonzero when graceful restart is enabled.

When you disable graceful restart but enable helper mode, the reconnect timeout is set to zero to announce to the neighbors that the LSR does not preserve MPLS forwarding state across the restart. The presence of the TLV indicates to the neighbors that the LSR supports them if they gracefully restart. That is, the LSR in this case waits for a gracefully restarting neighbor to resume sending messages.

Table 26 summarizes the states possible for LDP graceful restart.

Table 26: Summary of LDP Graceful Restart States

Graceful restart enabled?	Helper mode enabled?	FT TLV sent to neighbor?	Reconnect timeout value sent in TLV
Yes	Yes	Yes	Nonzero
No	Yes	Yes	Zero
No	No	No	–

The recovery time specifies how long the LSR retains its MPLS forwarding state across the restart. When the LSR restarts, it marks the forwarding state entries as stale. The forwarding state holding timer begins counting down from the configured recovery time value. If the timer expires before the restart completes, the LSR deletes all stale entries. When the LSR sends new LDP initialization messages to its neighbors, the messages contain the current value of the timer.

When the LSR restarts, if a neighbor of the LSR has previously received the FT session TLV from the LSR with a nonzero reconnect timeout value, the neighbor retains the label mapping information that it has previously received from the LSR and marks that information as stale. Alternatively, if the neighbor received an FT session TLV with a timeout value of zero (indicating that only helper mode is enabled) or no TLV at all (indicating that both graceful restart and helper mode are disabled), it deletes the label mapping information.

Also when the LSR restarts, the neighbor sets its neighbor liveness timer to the lesser of the two values, the reconnect timeout value and its own configurable neighbor liveness timer value. If the neighbor liveness timer expires, the neighbor deletes all the stale mappings from the LSR. The configurable value represents the maximum time that the neighbor waits for the restarting LSR to reestablish the LDP session. This enables the neighbor to avoid having to wait an unreasonably long time set by the reconnect timeout value from the restarting LSR.

If the recovery time value in the FT session TLV is zero when a neighbor receives the new LDP initialization message, the neighbor deletes all the stale mappings from the LSR.

If the recovery time value is nonzero, the neighbor starts a neighbor recovery timer set to the lesser of the two values, the recovery time value and its own configurable maximum recovery timeout value. The neighbor also cancels its neighbor liveness timer because the LDP session has been reestablished; it is now waiting on the successful completion of the restart.

The restarting LSR and its neighbors then exchange label mapping information. When a neighbor receives a label–to–FEC binding that matches a stale entry, it removes the staleness marker from the entry. If instead the neighbor receives a new label for the same FEC that is in a stale entry, the neighbor updates the entry with the new label and removes the staleness marker from the entry.

The neighbor deletes any stale entries that remain when the neighbor recovery timer expires.

Dynamic exchange of the graceful restart capability is not supported. In some circumstances, such as when a standby SRP module is removed, an LSR that has communicated to neighbors that it supports graceful restart might subsequently be unable to do so. In such cases, the neighbors receive no indication of that change in support unless you bounce the LDP sessions, for example by issuing the **clear mpls ldp neighbor** command.

mpls ldp graceful-restart

- Use to enable LDP graceful restart and helper mode. LDP graceful restart and helper mode are both disabled by default.
- Graceful restart causes the fault tolerant session TLV to be included in LDP initialization messages with a nonzero value for the reconnect timeout. This action announces to neighbors that the router preserves its forwarding state across a restart.
- Helper mode is automatically enabled when you enable graceful restart. Issue the **helper** keyword only when graceful restart is disabled and you want to enable helper mode alone.
- Helper mode causes the fault tolerant session TLV to be included in LDP initialization messages. The reconnect timeout value in the TLV is zero when LDP graceful restart is disabled. This TLV announces to neighbors that the router preserves any label-FEC mapping it has received from a neighbor in the event that the neighbor performs an LDP graceful restart.
- Example

  host1(config)#**mpls ldp graceful-restart**

- Use the **no** version to disable both LDP graceful restart and helper mode. You cannot disable helper mode alone.

mpls ldp graceful-restart reconnect-time

- Use to specify the length of time, in seconds, you want the neighbors to wait for the gracefully restarting router to resume sending LDP messages to neighbors after the LDP connection between them fails.
- This value is included in the fault tolerant session TLV sent in LDP initialization messages when LDP graceful restart is enabled.
- Specify a number in the range 60–300.
- Example

  host1(config)#**mpls ldp graceful-restart reconnect-time 130**

- Use the **no** version to restore the default value, 120 seconds.

mpls ldp graceful-restart recovery-time

- Use to specify the length of time, in seconds, the router retains its MPLS forwarding state across a restart.
- Specify a number in the range 120–600.
- Example

  host1(config)#**mpls ldp graceful-restart recovery-time 150**

- Use the **no** version to restore the default value, 120 seconds.

mpls ldp graceful-restart timers max-recovery

- Use to specify the maximum length of time, in seconds, that the router waits for a neighbor to complete a graceful LDP restart after the LDP session is reestablished.
- Specify a number in the range 15–600.
- Example

  host1(config)#**mpls ldp graceful-restart timers max-recovery 150**

- Use the **no** version to restore the default value, 120 seconds.

mpls ldp graceful-restart timers neighbor-liveness

- Use to specify the length of time, in seconds, the router waits for a neighbor to reestablish the LDP session.
- Specify a number in the range 5–300.
- Example

  host1(config)#**mpls ldp graceful-restart timers neighbor-liveness 150**

- Use the **no** version to restore the default value, 120 seconds.

Configuring LDP Autoconfiguration

LDP autoconfiguration with the **mpls ldp autoconfig** command enables you to ensure that LDP is configured on all interfaces running the IGP (IS-IS or OSPFv2). Using this command prevents you from having to configure LDP individually on each interface in the IGP. You can prevent LDP from being enabled on selected interfaces by issuing the **no mpls ldp autoconfig** command on the interface.

When autoconfiguration is enabled for IS-IS, you can specify whether LDP is automatically configured on all IS-IS interfaces in the virtual router or just the interfaces in a particular IS-IS level. When autoconfiguration is enabled for OSPF, you can specify whether LDP is automatically configured on all OSPF interfaces in the virtual router or just the interfaces in a particular OSPF area.

mpls ldp autoconfig

- Use to create LDP on the current interface when you issue the command from Interface Configuration mode.
- When you issue the command from Router Configuration mode for IS-IS, use to create LDP on all interfaces (on all levels) in the current router instance or on all interfaces in the specified level.
- When you issue the command from Router Configuration mode for OSPFv2, use to create LDP on all interfaces (in all areas) in the current router instance or on all interfaces in the specified area.
- By default, LDP autoconfiguration is not configured for IS-IS or OSPFv2.
- Example 1

  host1(config)#**router ospf 1**
  host1(config-router)#**mpls ldp autoconfig area 1**

- Example 2

  host1(config)#**router isis 1**
  host1(config-router)#**mpls ldp autoconfig level 1**

- Example 3

  host1(config)#**interface atm 2/0**
  host1(config-if)#**mpls ldp isis autoconfig**

- Use the **no** version from Interface Configuration mode to remove LDP from the interface. Use the **no** version from Router Configuration mode to remove the LDP configuration from the interfaces on which it was configured in that router instance.

Configuring LDP-IGP Synchronization

LDP is often used to establish MPLS LSPs throughout a complete network domain using an IGP such as OSPFv2 or IS-IS. In such a network, all links in the domain have IGP adjacencies as well as LDP adjacencies. LDP establishes the LSPs on the shortest path to a destination as determined by IP forwarding.

MPLS data packets can be discarded in these networks when the network IGP is operational on a link for which LDP is not fully operational, because there is no coupling between the LDP operational state and the IGP. When LDP is not fully operational, LDP is considered to not be synchronized with the IGP.

This issue is especially significant for applications such as a core network that does not employ BGP. Another example is an MPLS VPN where each given PE router depends on the availability of a complete MPLS forwarding path to the other PE routers for each VPN that it serves. This means that along the shortest path between the PE routers, each link must have an operational hello adjacency and an operational LDP session, and MPLS label bindings must have been exchanged over each session.

When LDP has not completed exchanging label bindings with an IGP next hop, traffic is discarded if the head end of the LSP forwards traffic because the LSP is assumed to be in place. The following are some examples of when this can happen.

- When an LDP hello adjacency or an LDP session with a peer is lost due to some error while the IGP still points to that peer. IP forwarding of traffic continues on the IGP link associated with the LDP peer rather than being shifted to another IGP link with which LDP is synchronized.

- When a new IGP link comes up, causing the next hop to a certain destination to change in the IGP's SPF calculations. Although the IGP might be up on the new link, LDP might not have completed label exchange for all the routes. This condition might be transient or due to a misconfiguration.

The LDP protocol is unable to indicate to a dependent service the availability of an uninterrupted LSP to the desired destination. LDP-IGP synchronization minimizes this disruption due to LDP not being operational on a link for which the IGP is operational. When synchronization is in effect, the IGP advertises the maximum possible cost or metric for that link. If an alternative next hop exists for traffic, the IGP can choose that next hop for forwarding. If LDP is operational for that next hop, then no traffic is discarded.

The IGP does not advertise the original cost or metric for the link until either LDP label exchange has been completed with the peer on the link or a configured amount of time has passed (the holddown period).

With synchronization configured, LDP notifies the IGP to advertise the maximum cost for the link when one of the following triggering events takes place:

- The LDP hello adjacency goes down.
- The LDP session goes down.
- LDP is configured on an interface.

If the holddown timer has been configured, the timer starts when the triggering event takes place. When the timer expires, LDP notifies the IGP to resume advertising the original cost.

If the holddown timer has not been configured, the IGP waits (endlessly) until bindings have been received from downstream routers for all the FECs that have a next hop on that interface. Only after that takes place does LDP notify the IGP to bring down the cost on the interface.

LDP-IGP synchronization is supported only for directly connected peers and links with the platform label space.

mpls ldp igp sync holddown

- Use to configure a holddown timer, in milliseconds, for LDP-IGP synchronization. This timer limits how long the IGP waits to synchronize with LDP.
- By default, the IGP waits indefinitely for LDP to be operational on the interface.
- Example

  host1(config)#**mpls ldp igp sync holddown 15**

- Use to configure a holddown timer for LDP-IGP synchronization.
- Use the **no** version to restore the default condition.

mpls ldp sync

- Use to synchronize LDP with the IGP on the current interface when you issue the command from Interface Configuration mode.
- From Router Configuration mode, use to synchronize LDP with the IGP on all interfaces in the current protocol router instance, on all interfaces in the specified level (IS-IS), or on all interfaces in the specified area (OSPFv2).

- By default, LDP synchronization is not configured.
- LDP-IGP synchronization must be configured on both sides of a link to be effective and avoid asymmetric link costs. In addition, LDP-IGP synchronization is effective only when alternate links with adequate bandwidth are available in the network.
- Example 1

  host1(config)#**router ospf 1**
  host1(config-router)#**mpls ldp ospf sync**

- Example 2

  host1(config)#**interface atm 2/0**
  host1(config-if)#**mpls ldp isis sync**

- Use the **no** version from Interface Configuration mode to stop LDP synchronization on the interface. Use the **no** version from Router Configuration mode to stop LDP synchronization on the interfaces on which it was configured in that router instance.

Synchronization Behavior During Graceful Restart

LDP-IGP synchronization does not take place while the IGP is in the process of a graceful restart. When the graceful restart completes, links for which synchronization has been configured are advertised with maximum metrics in either of the following cases:

- LDP is not yet operational on the link and no holddown timer has been configured.
- The configured holddown timer has not expired.

During LDP graceful restart, no synchronization operations are done. If the LDP graceful restart is terminated, LDP notifies the IGPs to advertise the links with the maximum metric.

Synchronization Behavior on LAN Interfaces

LDP-IGP synchronization does not take place on LAN interfaces unless the IGP has a point-to-point connection over the LAN configured on the interface. The reason for this is that multiple LDP peers might be connected on such an interface unless a point-to-point connection to a single peer has been configured. Because synchronization raises the cost on the interface high enough to prevent traffic being forwarded to that link, if multiple peers are connected, the cost is raised on all the peers even though LDP might be unsynchronized with only one of the peers. Consequently, traffic is diverted away from all the peers, an undesirable situation.

Synchronization Behavior on IGP Passive Interfaces

On IGP passive interfaces, the link cost is not raised when LDP-IGP synchronization is configured and a triggering event occurs.

Synchronization and TE Metrics

When traffic engineering is configured for an IGP, LDP-IGP synchronization does not affect the traffic engineering metric advertised for the link, regardless of whether the TE metric is explicitly configured or the default value.

Configuring LDP MD5 Authentication

LDP MD5 authentication provides protection against spoofed TCP segments that can be introduced into the connection streams for LDP sessions. Authentication is configurable for both directly connected and targeted peers.

You configure a shared secret (password) on potential LDP peers. Any given pair of peers must share the same password. When a peer sends a TCP segment to an LSR, it uses the password and the segment to compute an MD5 digest that it sends along with the segment.

When the LSR receives the segment, the LSR calculates its own version of the digest using its instance of the password and the segment. The LSR validates the segment if the local digest matches the received digest. If the comparison fails—for example, if the password is not configured the same on both peers—the LSR drops the segment and does not send a response to the peer.

You can optionally enable a strict authentication mode that allows only peers configured with passwords to establish sessions. In this mode, LDP hello messages from peers that have no password are ignored. If you do not configure strict authentication, then peers that do not have configured passwords can establish connections with each other.

If you configure LDP MD5 authentication or change the authentication password for a peer while it is in an established LDP session, MPLS restarts that session.

mpls ldp neighbor password

- Use to set a password for the specified LDP peer. MPLS uses the password to compute the MD5 digest for the peer for comparison with the MD5 digest sent by the remote peer when it attempts to establish a TCP connection.
- Example

  host1(config)#**mpls ldp neighbor 10.3.5.1 password rop23ers**

- Use the **no** version to delete the password.

mpls ldp strict-security

- Use to set strict LDP authentication mode. In this mode, only those peers that have passwords configured can establish sessions.
- Example

  host1(config)#**mpls ldp strict-security**

- Use the **no** version to remove the strict requirement and enable peers without configured passwords to establish connections with each other.

Configuring RSVP MD5 Authentication

RSVP MD5 authentication provides hop-by-hop security against message spoofing and replay attacks. When authentication is configured, RSVP embeds an integrity object within secure cleartext RSVP messages sent between peers. The integrity object includes a key ID unique to the sender, a message sequence number, and keyed message digest. These attributes enable verification of both packet content and sender.

For all potential RSVP peers, you configure the same key on the MPLS neighbor major interfaces, and then enable RSVP authentication on each of these interfaces. When you enable RSVP authentication on an interface, RSVP creates a security association that includes the key, key ID, hash algorithm, and other associated attributes. Each sender and receiver pair maintains the security association for their shared key.

NOTE: You must enable authentication on both ends of an RSVP interface to protect the link. Failure to do so can prevent tunnels through the interface from coming up.

Thereafter, RSVP messages sent by a router through the secured interface include an integrity object that contains a key ID for the security association and an MD5 message digest of the message contents. To protect against message replay attacks, the sending interface also places a sequence number in the integrity object. Each sequence number is a unique, monotonically increasing number.

The secured interface expects each received RSVP message to include an integrity object. The interface drops all RSVP messages that do not contain the object.

The receiver uses the key ID and the sender's address to determine the relevant security association. The key ID is extracted from the received integrity object. The address of the sending interface is extracted from the rsvp_hop object, if present, or from the packet header if the message does not include the rsvp_hop object. The receiver then recomputes the message digest using the association key and algorithm and compares it to the digest received from the peer.

If the digests match, RSVP checks the received sequence number. Every message received from a sender after the first authenticated message must have a sequence number greater than the number from a previously authenticated message from that sender. Messages with invalid sequence numbers are discarded.

If the sequence number is valid, then the RSVP message is authenticated and forwarded for normal RSVP processing. Unauthenticated messages are discarded.

clear rsvp authentication

- Use to clear the security association on a receiving peer for the specified sending peer.
- Use the **ip** keyword instead of **mpls** for compatibility with non–E-series implementations.

- Example

  host1#**clear mpls rsvp authentication 10.3.5.1**

- There is no **no** version.

mpls rsvp authentication

- Use to enable MD5 authentication on the RSVP interface after you have configured an authentication key.
- The router generates an error message and discards any RSVP messages if you enable authentication before configuring the authentication key, or remove the key while authentication is still enabled.
- Use the **ip** keyword instead of **mpls** for compatibility with non–E-series implementations.
- Example

  host1(config-if)#**mpls rsvp authentication**

- Use the **no** version to disable authentication on the interface.

mpls rsvp authentication key

- Use to assign a key to the interface for MD5 authentication between RSVP peers.
- Assign the authentication key before you enable authentication on the interface.
- Use the **ip** keyword instead of **mpls** for compatibility with non–E-series implementations.
- Example

  host1(config-if)#**mpls rsvp authentication key 34udR973j**

- Use the **no** version to delete the authentication key.

Failure Protection with RSVP-TE Bypass Tunnels

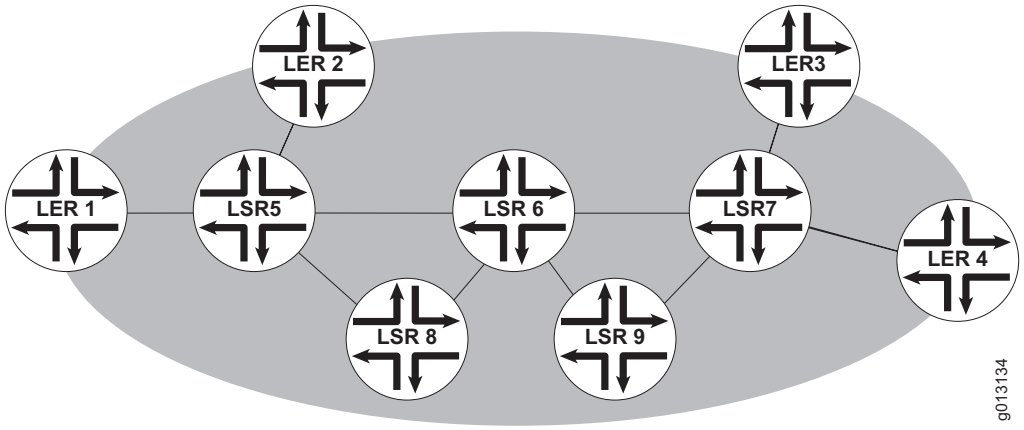
The fast reroute extensions to RSVP-TE enable you to create a single LSP, known as a bypass tunnel, to back up a set of LSPs by bypassing specific links in the LSP. In the event of a failure in any link of the protected RSVP-TE LSP (the primary LSP), MPLS redirects traffic to the associated bypass tunnel in tens of milliseconds.

You must statically configure the bypass tunnel for each link that you want to protect on each router in the LSP. The bypass tunnel must intersect the protected LSP at two locations. The start of the bypass tunnels is the point of local repair (PLR), which is the head end of the protected link. The bypass tunnel terminates downstream of the PLR on the node that represents the end of the bypassed link on the primary LSP.

Each bypass tunnel provides 1:*N* local protection; that is, each bypass tunnel can protect one or more links depending on where you have configured it. The protected primary LSPs are stacked over the bypass tunnel to redirect their traffic around the failure.

The bypass tunnel naturally protects all LSPs that share the bypassed link (the LSP segment from the PLR to the downstream node) and that have requested protection. Consider the network shown in Figure 60.

Figure 60: Bypass Tunnel



Suppose you have configured both LER 1 and LER 2 to request bypass protection, and have configured the following two bypass tunnels:

LSR 5 –> LSR 8 –> LSR 6
LSR 6 –> LSR 9 –> LSR 7

If link LSR 5 –> LSR 6 fails, RSVP-TE redirects traffic through LSR 5 –> LSR 8 –> LSR 6. If link LSR 6 –> LSR 7 fails, RSVP-TE redirects traffic through LSR 6 –> LSR 9 –> LSR 7. These two bypass tunnels therefore protect all LSPs routed from LER 1 or LER 2 through LSRs 5, 6, and 7. Notice in Figure 60 that if both protected links fail, traffic is still safely redirected through LSR 5 –> LSR 8 –> LSR 6 –> LSR 9 –> LSR 7.

If you want to protect an LSP that traverses *N* nodes against a failure in any link, then you must configure *N*-1 bypass tunnels. As shown in Figure 60, each of those bypass tunnels in turn can protect multiple tunnels.

On detecting the link failure, the PLR redirects traffic arriving on all of the protected primary tunnels to the bypass tunnel that protects the failed link. An additional label representing the bypass tunnel is stacked on the redirected packets. This label is popped either at the router that is the remote end of the protected link or at the penultimate hop. The merge point therefore sees traffic with the original label representing the primary tunnel.

When the ingress router learns by RSVP-TE signaling that local protection (a bypass tunnel) is in use, it attempts to find a new optimal path for the tunnel, based on the configured path options. The ingress router sets up the new tunnel before it tears down the old tunnel with the failed link, and switches its traffic to the new tunnel.

You can use the **tunnel mpls path-option** command to configure path options on the bypass tunnel. However, the link being protected by the bypass tunnel must not be in the path if you specify an explicit path.

Configuration Example

The following steps show a partial configuration using the topology in Figure 60.

1. On LSR 5, create a bypass tunnel to protect the link between LSR 5 and LSR 6. If you configure path options, you can specify the explicit path for the bypass tunnel either statically or to be dynamically calculated by the IGP traffic engineering mechanism.

   host1(config)#**interface tunnel mpls:bypass56**
   host1(config-if)#**tunnel mpls path-option 1 explicit name bypass**
   host1(config-if)#**tunnel destination 172.20.1.1**
   host1(config-if)#**exit**

2. On LSR 5, enable the explicit path, if configured.

   host1(config)#**mpls explicit-path name bypass enable**
   host1(config-expl-path)#**next-address 10.10.9.2**
   host1(config-expl-path)#**exit**

3. On LSR 5, assign the bypass tunnel to the interface being protected.

   host1(config)#**interface atm 4/0.1**
   host1(config-if)#**mpls backup-path bypass56**

4. On LER 1 (the tunnel ingress), specify that local protection is required for the primary tunnel.

   host1(config)#**interface tunnel mpls:primary1**
   host1(config-if)#**tunnel mpls fast-reroute**

mpls backup-path

- Use to assign the specified bypass tunnel to the interface that you want to protect.
- Example

  host1(config-if)#**mpls backup-path bypass1**

- Use the **no** version to remove the assignment.

tunnel mpls fast-reroute

- Use to configure local protection for the ingress router of the primary LSP.
- At setup of an LSP from this ingress router, RSVP-TE signals that the primary LSP needs local protection.
- Example

  host1(config-if)#**tunnel mpls fast-reroute**

- Use the **no** version to remove the configuration.

Fast Reroute over SONET/SDH

If you are using MPLS fast reroute over a SONET/SDH interface, reduce the times that the router uses to convert a defect to an alarm. Use the **path trigger delay** command to reduce the time for the path layer and the **trigger delay** command to reduce the time for the section and line layers. Use the following guidelines to set the times:

- Specify a value of 0 milliseconds if the interface does not use APS/MSP or if you want MPLS to have priority over APS/MSP.
- Specify a value of at least 100 milliseconds if this interface uses APS/MSP and if you want APS/MSP to have priority over MPLS.

For more information about these commands, see *JUNOSe Physical Layer Configuration Guide, Chapter 3, Configuring Unchannelized OCx/STMx Interfaces*, and *JUNOSe Physical Layer Configuration Guide, Chapter 4, Configuring Channelized OCx/STMx Interfaces*.

Determining Peer Reachability with RSVP-TE Hello Messages

RSVP-TE hello messages enable the router to detect when an RSVP-TE peer is no longer reachable. When the router makes this determination, all LSPs that traverse that neighbor are torn down. Hello messages are optional and can be ignored safely by peers that are not configured to use the feature.

When you enable the hello feature on a virtual router or interface configured with RSVP-TE, that RSVP-TE node periodically sends a unicast hello message to each neighbor with which the node has established an LSP. The exchange of hello messages between the peers establishes a hello adjacency. You can configure the hello interval to establish how frequently the node sends hello messages. Hello messages are exchanged when an LSP is set up and are stopped when the last LSP between the two peers goes away.

You can use the hello feature to reduce the impact of RSVP-TE on system resources. Because a hello timeout is treated as a link failure, RSVP-TE can use the hello timeout instead of path and resv timeouts to determine when to bring down an LSP.

You can increase the RSVP-TE refresh values to a large value to reduce the impact of RSVP-TE on system resources. High refresh values reduce the amount of control traffic (and CPU cycles) needed by RSVP-TE to refresh LSP state across the network:

- A hello refresh interval of 1000 milliseconds (a rate of one hello every second) is appropriate for a small configuration—tens of interfaces—without causing performance degradation.
- For larger configurations, the default hello refresh interval of 10,000 milliseconds (a rate of one hello every 10 seconds) is more appropriate and typically does not cause performance degradation.

Hello Message Objects

Hello messages can contain a hello request object or a hello ack object. These objects provide a way to request an instance value from a peer and to provide an instance value to a peer. Hello requests are sent to establish and confirm an adjacency with a peer. Hello acks are sent in response to hello requests. However, a hello adjacency peer can treat a hello request as an implicit response to its own request, thus reducing the amount of polling that needs to be done for efficient failure detection.

Hello Message Instances

Each object includes a source instance and a destination instance. The sender generates the source instance for its relationship with the receiver. The value of the source instance is unique for each peer. A given source instance value does not change while the two peers are successfully exchanging hello messages.

The destination instance is simply the source instance value that an RSVP-TE node has most recently received from its peer. If the node has never received a hello message from that peer, then it sets the destination instance value to zero.

Hello adjacency peers monitor the source instance sent by their neighbors. When a peer detects that the value has changed or that its neighbor does not properly report the source instance that the peer transmitted, then the peer treats that neighbor as if it has reset. In these cases, the local peer changes the instance value that it advertises to the neighbor.

Sequence of Hello Message Exchange

When a peer receives a hello message with a hello request object, the receiver generates a hello message with a hello ack object. If the receiver has never received a hello from the sender and the source instance is nonzero, then the receiver updates the destination instance that it sends in response with this new value. When the original sender first receives a hello ack from the peer in response to the hello request, the sender updates the destination instance that it sends in the subsequent hello request with the nonzero source instance it receives in the hello ack.

Consider the following example. An LSP has been established between peers A and B. These adjacent peers have not yet exchanged hello messages.

1. Peer A sends a hello request to Peer B. The request object contains the following:
   - Source instance = 5 (generated by Peer A for this adjacency)
   - Destination instance = 0 (because it has never exchanged messages with Peer B)

2. Peer B receives the hello request and sends a hello ack to Peer A. The ack object contains the following:

   - Source instance = 8 (generated by Peer B for this adjacency)
   - Destination instance = 5 (because that is what Peer B detected in the source instance from peer A)

3. Peer A receives the hello ack and sends another hello request to peer B. The request object contains the following:

   - Source instance = 5 (generated by Peer A for this adjacency)
   - Destination instance = 8 (the source instance generated by Peer B for this adjacency)

The two peers continue exchanging hello messages until the LSP is torn down. The following is true for these message exchanges unless a peer resets:

- Peer A always sends source instance = 5 and destination instance = 8 to Peer B.
- Peer B always sends instance = 8 and destination instance = 5 to Peer A.

Determination That a Peer Has Reset

After the initial exchange of hello messages, both peers perform checks on the messages they receive to determine whether the peer has reset.

Behavior of the Requesting Peer

The requesting peer examines the ack messages it receives. It compares the source instance in each subsequent ack message with the previous value. If the value differs or is set to zero, then the requesting peer treats the acknowledging peer as if communication has been lost.

The requesting peer also determines whether the acknowledging peer is reflecting back the requesting peer's source instance. If the acknowledging peer advertises a wrong value in the destination instance field of the ack message, then the requesting peer treats the acknowledging peer as if communication has been lost.

Behavior of the Acknowledging Peer

The acknowledging peer examines the request messages it receives. It compares the source instance in each subsequent request message with the previous value. If the value differs or is set to zero, then the acknowledging peer treats the requesting peer as if communication has been lost.

The acknowledging peer also determines whether the requesting peer is reflecting back the acknowledging peer's source instance. It compares the destination instance value in each request message with the source instance value that it most recently advertised to the requesting peer. If the requesting peer advertises a wrong value in the destination instance field of the request message, then the acknowledging peer treats the requesting peer as if communication has been lost.

Behavior of Both Peers

When no hello messages are received from a peer within the configured hello interval, the peer is treated as if communication has been lost.

When a peer determines that communication has been lost, it can reinitiate the sending of hello messages. In this case, the peer generates a new source instance different than the one it previously used for communication with its peer.

mpls rsvp signalling hello

- Use in Global Configuration mode to enable or configure RSVP-TE hellos for all RSVP-TE interfaces on the current virtual router. By default, RSVP-TE hellos are disabled on the virtual router.

  Issuing this command in Global Configuration mode automatically creates RSVP-TE or MPLS on the current virtual router if they do not yet exist there.

- Use in Interface Configuration mode to enable or configure RSVP-TE hellos for the current RSVP-TE interface on the current virtual router. By default, RSVP-TE hellos are disabled on interfaces.

  Issuing this command in Interface Configuration mode automatically creates RSVP-TE or MPLS on the current virtual router if they do not yet exist there. This command also creates RSVP-TE on the interface if it is not configured there.

- Interfaces within a virtual router inherit the global configuration settings. Issuing this command in Interface Configuration mode overrides the global configuration for the current interface.
- Issuing the **refresh interval** or the **refresh misses** keywords only configures the refresh values; this action has no effect on enabling or disabling hellos.
  - Use the **refresh interval** keywords to specify the RSVP-TE hello interval on the current layer 2 interface or on all layer 2 interfaces in the virtual router. The default value is 10,000 milliseconds.
  - Use the **refresh misses** keywords to specify the number of missed RSVP-TE hellos required to declare the RSVP-TE neighbor down on the current layer 2 interface or on all layer 2 interfaces in the virtual router. The default value is 4 misses.
- Example 1—Enables RSVP-TE hellos on virtual router VR5; creates RSVP-TE, MPLS, or both on VR5 if not already present

  host1:vr5(config)#**mpls rsvp signalling hello**

- Example 2—Enables RSVP-TE hellos on FastEthernet 4/3; creates RSVP-TE, MPLS, or both on VR5 and creates RSVP-TE on the interface if not already present

  host1:vr5(config)#**interface fastEthernet 4/3**
  host1:vr5(config-if)#**mpls rsvp signalling hello**

- Example 3—Configures the refresh hello interval on FastEthernet 4/3; creates RSVP-TE, MPLS, or both on the default virtual router and creates RSVP-TE on the interface if not already present

  host1(config)#**interface fastEthernet 4/3**
  host1(config-if)#**mpls rsvp signalling hello refresh interval 5000**

- Example 4—Configures the refresh hello misses on FastEthernet 4/3; creates RSVP-TE, MPLS, or both on the default virtual router and creates RSVP-TE on the interface if not already present

  host1(config)#**interface fastEthernet 4/3**
  host1(config-if)#**mpls rsvp signalling hello refresh misses 3**

- Use the **no** version to disable RSVP-TE hellos on the current VR or on the current interface. Use the **default** version to restore the defaults: hellos are disabled, the hello interval is set to 10000 milliseconds, and hello misses are set to 4. In Interface Configuration mode, the **default** version also restores inheritance of the configuration values from the global configuration.

RSVP-TE Graceful Restart

RSVP-TE graceful restart enables routers to maintain MPLS forwarding state when a link or node failure occurs. In a link failure, control communication is lost between two nodes, but the nodes do not lose their control or forwarding state.

A node failure occurs when the LSR has a failure in the RSVP-TE control plane, but not in the data plane. The LSR maintains its data forwarding state. Traffic can continue to be forwarded while RSVP-TE restarts and recovers. The graceful restart feature supports the restoration and resynchronization of RSVP-TE states and MPLS forwarding state between the restarting router and its RSVP-TE peers during the graceful restart recovery period.

The RSVP-TE graceful restart feature enables an LSR to gracefully restart, to act as a graceful restart helper node for a neighboring router that is restarting, or both.

Announcement of the Graceful Restart Capability

LSRs use the RSVP-TE hello mechanism to announce their graceful restart capabilities to their peer RSVP-TE routers. Both restarting LSRs and helper LSRs include the restart_cap object in hello requests and hello acks. The restart_cap object specifies both the graceful restart time and the graceful restart recovery time:

- restart time—The sum of how long it takes the sender to restart RSVP-TE after a control plane failure plus how long it takes to reestablish hello communication with the neighboring RSVP-TE routers.

- recovery time—The period within which you want neighboring routers to resynchronize with the sending router's RSVP-TE state and MPLS forwarding state after the peers have re-established hello communication. The restarting LSR advertises the configured or default recovery time only while the graceful restart is in progress. When the LSR is not currently restarting or when it is incapable of preserving its MPLS forwarding state during the restart, the LSR advertises a recovery time of zero.

Both the restarting router and neighboring GR helper routers save the restart and recovery times that they receive from their peers.

Restarting Behavior

When the control plane fails, the LSR stops sending hello messages to its RSVP-TE neighbors. However, as a restarting router the LSR can continue to forward MPLS traffic because it preserves its MPLS forwarding state during the restart. When RSVP-TE comes back up, the restarted router sends the first hello message to its neighbors with a new source instance value to indicate that it had a control plane failure. The destination instance value in the hello message is set to zero. The recovery time included in this hello message is set to zero only if the router was unable to preserve the MPLS forwarding state or to support control state recovery.

When a neighboring router that has been configured as a graceful restart helper determines that the number of continuous missing hellos has reached the configured hello miss limit, it declares the router to be down. The helper router then waits for a period equal to the restart time that it received from the router and stored before the failure. During this period, the helper router preserves the restarting router's RSVP-TE state and MPLS forwarding state for the established LSPs and keeps forwarding MPLS traffic. However, the helper router suspends the refreshing of path and resv state to the restarting router. The helper router keeps sending hello messages to the restarting router with an unchanged source instance value and a destination instance value set to zero. The helper router removes the RSVP-TE state for any LSP that was in the process of being established when the neighbor was declared to be down.

If the helper router does not receive a hello message from the restarting router during the restart period, the helper router immediately exits the recovery procedure and cleans up the states associated with the restarting router. The helper router determines that the failed neighbor has restarted when it finds a new source instance in the neighbor's hello message. When a nonzero recovery time is received in that hello message, the helper router determines that the restarted neighbor supports state recovery. The helper router then starts the recovery procedures. However, if the recovery time specified in the hello message is zero, then the helper router exits the recovery procedure and cleans up the states associated with the restarting router.

Recovery Behavior

In the recovery period, neighboring helper routers and the restarting router resynchronize the RSVP-TE state and MPLS forwarding state. During this period, MPLS traffic continues to be forwarded.

The helper router starts the recovery procedure by marking as stale the RSVP-TE state associated with the restarting router. The upstream helper router then refreshes all the path messages shared with the downstream restarting router. The upstream helper router includes the recovery_label object in the path message to the downstream restarting router for the label binding information that the restarting router specified before the restart. The downstream helper router does not refresh the reservation state control block (RSB) shared with the restarting router until a corresponding path message is received from the restarting router.

During the recovery period, the restarting router checks for the state associated with an incoming path message. If the RSVP-TE state already exists, the restarting router handles the path message as usual. Otherwise, the restarting router examines the path message for the recovery_label object. If the recovery_label object is not found, the restarting router treats the path message as a setup request for a new LSP and handles the path message as usual.

If the recovery_label object is found, the restarting router searches for the outgoing label based on the incoming interface and incoming label that are specified in the recovery_label object. If the restarting router does not find a match for the forwarding entry, the restarting router treats the path message as a setup request for a new LSP. If the restarting router finds a match, it conveys to the downstream neighbors the outgoing label associated with the forwarding entry in the suggested_label object in the path message and it continues normal operations.

The helper router removes the stale flag for the RSVP-TE state when it receives the corresponding state in path or resv messages sent by the restarting router. When the recovery period expires, the helper router deletes any RSVP-TE states that still have a stale flag. Graceful restart is considered to be complete when the recovery period expires or when the last LSP needing recovery is recovered.

Preservation of an Established LSP Label

Labels used for an established LSP are preserved through the graceful restart by means of the recovery_label object and the suggested_label object in the path messages. The recovery_label object conveys the incoming label of the restarting LSR that the restarting LSR passed to the upstream helper before the restart. The suggested_label object includes the outgoing label that the restarting LSR used before the restart. The suggested_label object conveys the outgoing label from the restarting LSR to its downstream neighbor.

mpls rsvp signalling hello graceful-restart

- Use to enable RSVP-TE graceful restart on the current virtual router.
- Issue the **mode help-neighbor** keywords to restrict the VR to act only as a graceful restart helper node for neighbors that support RSVP-TE graceful restart. If you do not issue these keywords, the VR has full restarting node capability as well as graceful restart helper node capability.
- Issuing this command automatically creates RSVP-TE or MPLS on the current virtual router if they do not yet exist there.
- Example

  host1(config)#**mpls rsvp signalling hello graceful-restart**

- Use the **no** version to disable RSVP-TE graceful restart.

mpls rsvp signalling hello graceful-restart recovery-time

- Use to configure the time in, milliseconds, within which you want neighboring routers to resynchronize RSVP-TE state and MPLS forwarding state after a graceful restart.
- Issuing this command automatically creates RSVP-TE or MPLS on the current virtual router if they do not yet exist there.
- Specify a number in the range 60000–480000.
- Example

  host1(configf)#**mpls rsvp signalling hello graceful-restart recovery-time 140000**

- Use the **no** version to restore the default value, 120,000 ms.

mpls rsvp signalling hello graceful-restart restart-time

- Use to configure the time, in milliseconds, within which the sender gracefully restarts RSVP-TE and reestablishes hello communication with RSVP-TE neighbors.
- Issuing this command automatically creates RSVP-TE or MPLS on the current virtual router if they do not yet exist there.
- Specify a number in the range 60000–3600000.
- Example

  host1(configf)#**mpls rsvp signalling hello graceful-restart refresh restart-time 150000**

- Use the **no** version to restore the default value, 60,000 ms.

Using RSVP-TE Hellos Based on Node IDs

You can use the **mpls rsvp signalling node-hello** command to configure the exchange of node-ID–based RSVP-TE hellos (node hellos) for interoperability with routers that cannot support RSVP-TE graceful restart with link-based hellos. E-series routers use node hellos only to support their graceful restart capabilities.

NOTE: Node hellos are not required for RSVP-TE graceful restart support between routers running JUNOSe software or for interoperability with routers running JUNOS software.

Graceful restart must be enabled for node hellos to advertise graceful restart. Link-based hellos are not required for graceful restart when you have configured node hellos. However, you might still use link-based hellos to monitor RSVP-TE links and detect link failures.

The node hello sessions are established by the exchange of hello messages in which node IDs are used for the source and destination addresses in the hello packets. The sending router uses its local node ID as the source address and the remote node ID of the receiving router as the destination address.

RSVP-TE uses the configured IGP, IS-IS or OSPF, to learn the local and remote node IDs. In IS-IS, the node ID is the TE router ID as defined in the traffic engineering router ID TLV for IPv4 addresses and in the IPv6 TE Router_ID for IPv6 addresses. In OSPF, the node ID is the TE router ID as defined in the router address TLV for IPv4 addresses and in the Router_IPv6_Address for IPv6 addresses. Only one node-based RSVP-TE hello session can be established for each instance of an IGP adjacency with a peer.

When a router receives a hello message where the destination address is set to the receiving router's local node ID, the router verifies that the node ID is the ID that the IGP advertises. This router must then use its local node ID as the source address when it replies to the sending router.

Node-based hellos are an attractive alternative to link-based hellos for graceful restart when you use bidirectional forwarding detection (BFD) for link monitoring and you have configured node-based hellos on all RSVP-TE peers.

Link-based RSVP-TE hellos are used for monitoring RSVP-TE adjacencies with neighboring routers and for providing RSVP-TE graceful restart. However, the BFD protocol is more effective at monitoring RSVP-TE adjacencies than are link-based hellos.

Link-based RSVP-TE hellos for graceful restart are more resource-intensive option than node-based RSVP-TE hellos when your configuration has several interfaces enabled with MPLS RSVP-TE and carrying RSVP-TE data traffic. Link-based hellos generate a volume of network traffic and processing overhead that is directly proportional to the number of interfaces that are carrying active RSVP-TE tunnels.

Node-based hellos require less messaging and processing overhead in these circumstances. Node hellos require only a single hello session between the two node IDs, compared to link-based hellos that have hello sessions between all interface pairs. Less traffic and overhead result in a lesser impact on scaling.

Node-based hellos can therefore be advantageous even when you are interoperating with routers that are running JUNOSe software or JUNOS software, if you are using BFD to monitor your RSVP-TE links. If you are not using BFD, then you must use link-based hellos for link monitoring, and link-based hellos then become more practical for graceful restart.

mpls rsvp signalling node-hello

- Use in Global Configuration mode to enable or configure node-ID–based RSVP-TE hellos for all RSVP-TE interfaces on the current virtual router. By default, RSVP-TE node hellos are disabled on the virtual router.

  Issuing this command automatically creates RSVP-TE or MP534

  LS on the current virtual router if they do not yet exist there.

- Graceful restart must be enabled on the VR so that the node hellos can advertise the graceful restart capabilities.
- Issuing the **refresh interval** or the **refresh misses** keywords only configures the refresh values; this action has no effect on enabling or disabling RSVP-TE node hellos.
  - Use the **refresh interval** keywords to specify the RSVP-TE node hello interval on the current layer 2 interface or on all layer 2 interfaces in the virtual router. The default value is 10,000 milliseconds.
  - Use the **refresh misses** keywords to specify the number of missed RSVP-TE node hellos required to declare the RSVP-TE neighbor down on the current layer 2 interface or on all layer 2 interfaces in the virtual router. The default value is 4 misses.
- Example

  host1:vr5(config)#**mpls rsvp signalling hello graceful-restart**
  host1:vr5(config)#**mpls rsvp signalling node-hello**

- Use the **no** version to disable RSVP-TE node hellos on the current VR. Use the **default** version to restore the defaults: node hellos are disabled, the hello interval is set to 10000 milliseconds, and hello misses are set to 4.

Configuring the BFD Protocol for RSVP-TE

The **mpls rsvp bfd-liveness-detection** command configures the Bidirectional Forwarding Detection (BFD) protocol for RSVP-TE. The BFD protocol uses control packets and shorter detection time limits to more rapidly detect failures in a network. Also, because they are adjustable, you can modify the BFD timers for more or less aggressive failure detection.

Without BFD, RSVP-TE can learn about adjacency failures by either of two methods. If RSVP-TE hellos are configured, then hello message timeouts indicate a failure. If hellos are not configured, then RSVP-TE learns about failures from resv and path messages.

When a BFD session exists between RSVP-TE peers, a peer that goes down is detected quickly, enabling faster rerouting of traffic. Adjacency failure detection by means of hello messages takes place on the order of seconds, whereas BFD fast failure detection can take place on the order of hundreds of milliseconds.

When you issue the **mpls rsvp bfd-liveness-detection** command on an RSVP-TE major interface, BFD liveness detection is established with all BFD-enabled RSVP-TE peers associated with that interface.

When an RSVP-TE session is established with the remote peer—if BFD is enabled and if the BFD session is not already present—then the local peer attempts to create a BFD session to the remote peer. The BFD session is established only if when both of the following are true:

- At least one RSVP-TE LSP exists between (passes through) a pair of directly connected RSVP-TE major interfaces.
- Both interfaces are BFD-enabled.

Consequently, when the last LSP is torn down between the interfaces, the BFD session is no longer required and is brought down as well.

Each adjacent pair of peers negotiates an acceptable transmit interval for BFD packets. The negotiated value can be different on each peer. Each peer then calculates a BFD liveness detection interval. When a peer does not receive a BFD packet within the detection interval, it declares the BFD session to be down and purges all routes learned from the remote peer.

NOTE: Before the router can use the **mpls rsvp bfd-liveness-detection** command, you must specify a BFD license key. To view an already configured license, use the **show license bfd** command.

For general information about configuring and monitoring the BFD protocol, see *JUNOSe IP Services Configuration Guide, Chapter 5, Configuring BFD*.

mpls rsvp bfd-liveness-detection

- Use to enable BFD (bidirectional forwarding detection) and define BFD values on an RSVP-TE major interface to more quickly detect RSVP-TE data path failures.
- The peers in an RSVP-TE adjacency use the configured values to negotiate the actual transmit intervals for BFD packets.
  - You can use the the **minimum-transmit-interval** keyword to specify the interval at which the local peer proposes to transmit BFD control packets to the remote peer. Specify a number in the range 100–65535 milliseconds. The default value is 300 milliseconds.
  - You can use the **minimum-receive-interval** keyword to specify the minimum interval at which the local peer must receive BFD control packets from the remote peer. Specify a number in the range 100–65535 milliseconds. The default value is 300 milliseconds.
  - You can use the **minimum-interval** keyword to specify the same value for both of those intervals. Configuring a minimum interval has the same effect as configuring the minimum receive interval and the minimum transmit interval to the same value. Specify a number in the range 100–65535 milliseconds. The default value is 300 milliseconds.
  - You can use the **multiplier** keyword to specify the detection multiplier value. The calculated BFD liveness detection interval can be different on each peer. The multiplier value is roughly equivalent to the number of packets that can be missed before the BFD session is declared to be down. Specify a number in the range 1–255. The default value is 3.

- By default, BFD is not configured or enabled on RSVP-TE major interfaces.
- For details on liveness detection negotiation, see *Negotiation of the BFD Liveness Detection Interval* section in *JUNOSe IP Services Configuration Guide, Chapter 5, Configuring BFD*.
- You can change the BFD liveness detection parameters at any time without stopping or restarting the existing session; BFD automatically adjusts to the new parameter value. However, no changes to BFD parameters take place until the values resynchronize with each peer.
- Example

  host1(config-if)#**mpls rsvp bfd-liveness-detection minimum-interval 400**

- Use the **no** version to unconfigure BFD on the interface.

Verifying and Troubleshooting MPLS Connectivity

In IP networks, the **ping** and **traceroute** commands enable you to verify network connectivity and find broken links or loops. In MPLS-enabled networks, you can use the **ping** command to determine whether IP connectivity exists to a destination even when the ping packets must traverse multiple LSPs. You can use the **traceroute** command to determine the labels that data packets use when traversing LSPs to the destination.

In an MPLS-enabled network, however, you cannot use these IP commands to determine MPLS connectivity to a destination. Instead, you can use the MPLS ping and trace features to detect data plane failures in LSPs. Specific **mpls ping** and **trace mpls** commands enable you to target different types of MPLS applications and network topologies. The various **ping mpls** and **trace mpls** commands send UDP packets, known as MPLS echo requests, to the egress LSR of MPLS packets in a given FEC. Each echo request is forwarded along the same data path as the MPLS packets in that FEC.

The echo request packets use a destination address in the 127.0.0.0/8 range and port 3503. The default address is 127.0.0.1. This address range prevents IP from forwarding the packet, so that the echo request must follow the MPLS data path. This behavior is different from that of the IP **ping** and **traceroute** commands, which send ICMP packets to the actual destination.

Each MPLS echo request packet contains information about the FEC stack that is being validated. LSRs that receive an MPLS echo request respond with MPLS echo reply packets.

The **ping mpls** commands perform a basic connectivity check. When the echo request exits the tunnel at the egress LSR, the LSR sends the packet to the control plane. The egress router validates the FEC stack to determine whether that LSR is the actual egress for the FEC. The egress router sends an echo reply packet back to the source address of the echo request packet. The egress router can send the packet back by means of either the IP path or the MPLS path.

The **trace mpls** commands isolate faults in the LSP. For these commands, successive echo request packets are sent along the path. The first packet has a TTL of one; the TTL value is incremented by one for each successive packet. The first packet therefore reaches only the next hop on the path; the second packet reaches the next router after that. Echo request packets are sent until either an echo reply is received from the egress router for the FEC or a TTL of 32 is reached.

When a TTL expires on an LSR, that LSR sends an echo reply packet back to the source. For transit routers, the echo reply indicates that downstream mapping exists for the FEC, meaning that the packet would have been forwarded if the TTL had not expired. The egress router sends an echo reply packet verifying that it is the egress.

Although you cannot send IPv6 UDP packets for MPLS ping, you can use the **ping mpls l3vpn** command with an IPv6 prefix to investigate IPv6 VPNs.

MPLS Echo Reply Generation

Echo reply packets are sent by E-series routers that receive an echo request packet, even when MPLS is not enabled on that router. This situation is a transient condition when the router is receiving labeled packets. A return code in the echo replies indicates to the sending router that no label mapping exists on the receiving router.

MPLS Connectivity and ECMP

When an MPLS ECMP is part of the tunnel being explored by an MPLS echo request, the request packet takes one of the available ECMP paths. Probing FECs with different label stacks can yield different ECMP paths. However, you cannot guarantee complete coverage of all the ECMP paths.

You can use MPLS trace to determine which paths are present on an MPLS LSR. When the TTL expires on an MPLS LSR, the echo reply that is returned includes a downstream mapping TLV. This TLV contains all the downstream mappings of the LSR on which the TTL expired, if that feature is supported by the LSR. You can use the **detail** version of the **trace mpls** commands to display these downstream mappings.

Supported TLVs

Table 27 lists the TLVs supported by the MPLS LSP ping feature. Table 28 on page 287 lists the sub-TLVs supported for the Target FEC Stack TLV.

Table 27: TLVs Supported by MPLS LSP ping

Type Number	Value	Comments
1	Target FEC Stack	Multiple FEC stack sub-TLVs are not supported. A single LSP ping message cannot have more than one target FEC stack TLV.
2	Downstream Mapping	Only the IPv4 (numbered or unnumbered) downstream address type is supported. Flag I for the Interface and Label Stack object is supported. Flag N, to treat the packet as a non-IP packet, is not supported. An MPLS LSP trace echo request includes this TLV. This TLV contains the downstream address all-routers-multicast; that is the well-known IP address 224.0.0.2. Validation of the downstream address is not performed. Verification of the downstream address is not performed on receipt of an MPLS echo request that contains this TLV. In an MPLS echo reply, multipath information is not supported in this TLV; the multipath type is always set to 0 in the reply. However, the reply includes one downstream mapping TLV for each downstream path.
3	Pad	This TLV is included in the MPLS echo request packet. The TLV can specify either "Do not reply" or "Reply via an IPv4/IPv6 UDP packet."
7	Interface and Label Stack	This TLV is generated if requested by the received downstream mapping TLV.
9	Errored TLVs	This TLV is generated if an error is encountered while parsing one of the received TLVs.
10	Reply TOS Byte	–

Table 28: Sub-TLVs Supported for the Target FEC Stack TLV

Subtype Number	Value	Comments
1	LDP IPv4 prefix	–
2	LDP IPv6 prefix	–
3	RSVP IPv4 LSP	–
6	VPN IPv4 prefix	–
7	VPN IPv6 prefix	–
8	L2 VPN endpoint	For VPLS and L2VPN
10	FEC 128 pseudowire	For Martini encapsulation

ping mpls ip

- Use to send an MPLS echo request packet to the specified IP or IPv6 address.
- The MPLS echo request packets and echo reply packets created by this command use the LDP IPv4 LSP sub-TLV described in RFC 4379—Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures (February 2006).
- When you specify a VRF name, the LSP to the specified prefix must originate from the VRF because the ping is generated from the specified VRF.
- Example

  host1:pe1#**ping mpls ip 10.2.2.2/32**

- There is no **no** version.

ping mpls l2transport

- Use to send an MPLS echo request packet to the specified layer 2 cross-connect virtual (Martini) circuit.
- This command is not supported for local cross-connects because local cross-connects do not employ an LSP.
- The echo request packet generated by this command contains the FEC 128 Pseudowire (Current) sub-TLV described in RFC 4379—Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures (February 2006).
- If you specify a VRF name, the ping is generated from the specified VRF. For that reason the MPLS shim interface must exist in the VRF.
- By default, the TTL on the inner (stacked) label is set to 1 when transmitting echo request packets. This value causes the packet to be exceptioned to the SRP module when the stacked label is exposed.
- Example

  host1:pe1#**ping mpls l2transport FastEthernet1/0.1 detail**

- There is no **no** version.

ping mpls l3vpn

- Use to send an MPLS echo request packet to the specified L3VPN IP or IPv6 prefix.
- The echo request packet generated by this command contains either the VPN IPv4 sub-TLV or VPN IPv6 sub-TLV described in RFC 4379—Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures (February 2006). Which sub-TLV is included depends on whether the ping is intended for an IPv4 prefix or an IPv6 prefix.
- You can use this command to send a request to a VPNv4 prefix in the specified VRF. If you do not specify a VRF, then you must issue the command from the VRF context. In any case, the ping originates from the parent router.
- Example

  host1:pe1#**ping mpls l3vpn vrf pe11 10.32.45.21/32**

- There is no **no** version.

ping mpls rsvp tunnel

- Use to send an MPLS echo request packet to the specified RSVP-TE tunnel.
- The MPLS echo request packets and echo reply packets created by this command use the RSVP IPv4 sub-TLV described in RFC 4379—Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures (February 2006).
- Specify the VRF only when the RSVP-TE tunnel originates in the VRF because the ping is generated from the specified VRF.
- The tunnel specified with the **tunnel** keyword can be a bypass tunnel.
- Example

  host1:pe1#**ping mpls rsvp tunnel west1**

- There is no **no** version.

ping mpls vpls

- Use to send an MPLS echo request packet to the specified VPLS instance.
- The echo request packet generated by this command contains the layer 2 VPN endpoint sub-TLV described in RFC 4379—Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures (February 2006).
- You can specify a VRF context to generate the ping from the specified VRF.
- By default, the TTL on the inner (stacked) label is set to 1 while transmitting the echo request packet. This value causes the packet to be exceptioned to the SRP module when the stacked label is exposed.
- Example

  host1:pe1#**ping mpls vpls vrf pe11 vplsA remote-site-id 2**

- There is no **no** version.

trace mpls ip

- Use to send MPLS echo request packets to discover and examine the path MPLS packets follow to the specified IP or IPv6 address.
- The MPLS echo request packets and echo reply packets created by this command use the LDP IPv4 sub-TLV described in RFC 4379—Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures (February 2006).
- When you specify a VRF name, the LSP to the specified prefix must originate from the VRF because the ping is generated from the specified VRF.
- Example

  host1:pe1#**trace mpls ip 192.168.25.1/32**

- There is no **no** version.

trace mpls l2transport

- Use to send MPLS echo request packets to discover and examine the path MPLS packets follow to the specified layer 2 cross-connect virtual (Martini pseudowire) circuit.
- This command is not supported for local cross-connects because local cross-connects do not employ an LSP.
- The echo request packet generated by this command contains the FEC 128 Pseudowire (Current) sub-TLV described in RFC 4379—Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures (February 2006).
- By default, the TTL on the inner (stacked) label is set to 1 when transmitting echo request packets. This value causes the packet to be exceptioned to the SRP module when the stacked label is exposed.
- Example

  host1:pe1#**trace mpls l2transport FastEthernet1/0.1 detail**

- There is no **no** version.

trace mpls l3vpn

- Use to send MPLS echo request packets to discover and examine the path MPLS packets follow to the L3VPN IP or IPv6 prefix.
- The echo request packet generated by this command contains either the VPN IPv4 sub-TLV or VPN IPv6 sub-TLV described in RFC 4379—Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures (February 2006). Which sub-TLV is included depends on whether the trace is intended for an IPv4 prefix or an IPv6 prefix.
- You can use this command to send a request to a VPNv4 prefix in the specified VRF. If you do not specify a VRF, then you must issue the command from the VRF context. In either case, the trace originates from the parent router.
- Example

  host1:pe1#**trace mpls l3vpn vrf pe11 10.2.3.21/32**

- There is no **no** version.

trace mpls rsvp tunnel

- Use to send MPLS echo request packets to discover and examine the path MPLS packets follow to the specified RSVP-TE tunnel.
- The MPLS echo request packets and echo reply packets created by this command use the RSVP IPv4 sub-TLV described in RFC 4379—Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures (February 2006).
- Specify the VRF only when the RSVP-TE tunnel originates in the VRF because the ping is generated from the specified VRF.
- The tunnel specified with the **tunnel** keyword can be a bypass tunnel.

- Example

  host1:pe1:pe11#**trace mpls rsvp tunnel west1 detail**

- There is no **no** version.

trace mpls vpls

- Use to send MPLS echo request packets to discover and examine the path MPLS packets follow to the specified VPLS instance.
- The MPLS echo request packets and echo reply packets created by this command use the L2 endpoint sub-TLV described in RFC 4379—Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures (February 2006).
- When you specify a VRF name, the LSP to the specified prefix must originate from the VRF because the ping is generated from the specified VRF.
- By default, the TTL on the inner (stacked) label is set to 1 while transmitting the echo request packet. This value causes the packet to be exceptioned to the SRP module when the stacked label is exposed.
- Example

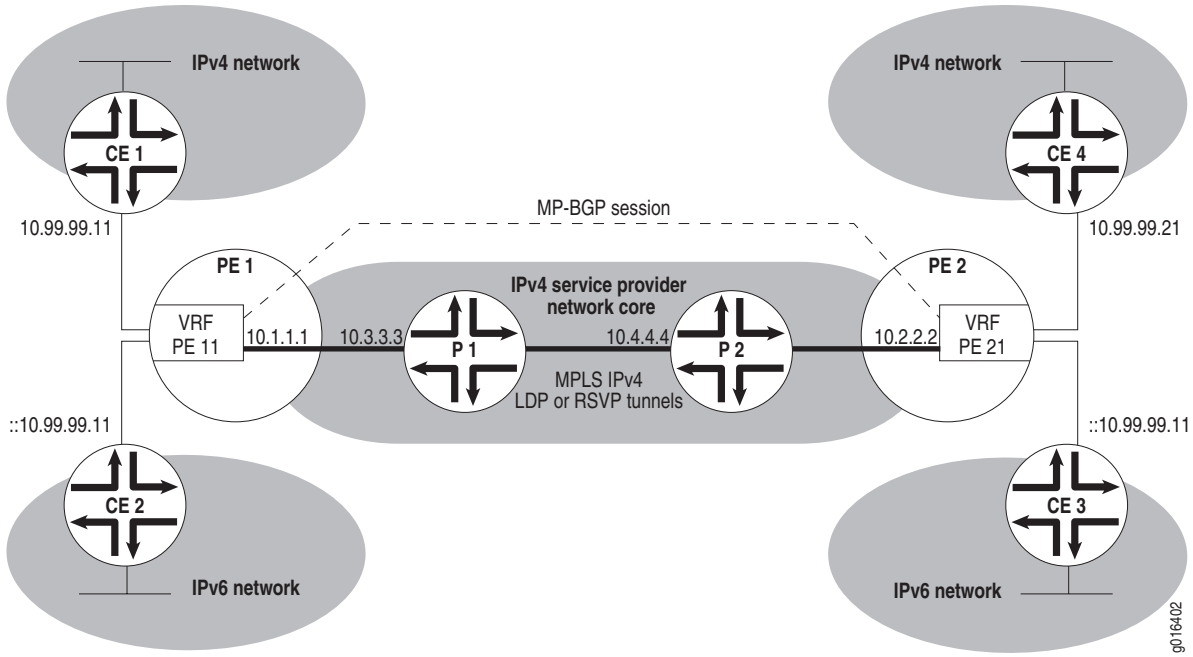
  host1:pe1#**trace mpls vpls vrf pe11 vplsA remote-site-id 2 detail**

- There is no **no** version.

Sample Network Topology

Figure 61 on page 292 shows a sample IPv4/IPv6 L3VPN topology with LDP or RSVP-TE base tunnels. Two base tunnels (one in each direction) are present between 10.1.1.1 and 10.2.2.2. The packet flow examples that follow refer to this sample topology.

Figure 61: Sample MPLS L3VPN Topology



MPLS LSPs to an IP prefix

Use the **ping mpls ip** and **trace mpls ip** commands for MPLS LSPs that are configured to use LDP; labeled BGP; or a combination of LDP, BGP, and RSVP-TE (as for inter-AS and carrier-of-carriers topologies). When you specify a VRF name, the LSP to the specified prefix must originate from the VRF because the ping is generated from the specified VRF.

Packet Flow Example for the ping mpls Command

The following example illustrates the packet flow that results when you issue the **ping mpls ip** command from router PE 1 (10.1.1.1) to router PE 2 (10.2.2.2) over an LDP base tunnel.

```
host1:pe1#ping mpls ip 10.2.2.2/32
```

1. PE 1 sends an MPLS echo request UDP packet that contains an LDP IPv4 sub-TLV. The packet is sent as a labeled packet over the target LSP. The packet has the following attributes:

Source address	10.1.1.1
Destination address	127.0.0.0/8
UDP port	3503
TTL	255
IPv4 prefix in the TLV	10.2.2.2/32
Sender's handle	Randomly generated 32-bit number used to match the reply
Sequence number	Integer that is incremented for each echo request packet

2. Router P 1 label-switches the packet to P 2.

3. Router P 2 label-switches the packet to PE 2 (assuming PHP is not configured).

4. Router PE 2 pops the label and determines that the destination address is in the 127.0.0.0/8 subnet. PE 2 sends the packet up to the control plane. The MPLS ping application on the control plane then creates an MPLS echo reply to the received echo request. The echo reply packet has a return code of 3, which means that the replying router is an egress for the FEC at stack depth. The echo reply packet includes the Interface and Label Stack TLV to indicate both the interface on which the request packet was received and the incoming label stack. The MPLS echo reply packet is sent back as a (labeled) UDP packet with the following attributes:

Source address	10.2.2.2
Destination address	10.1.1.1
UDP port	3503

5. When the MPLS echo reply reaches router PE 1, the router matches the sender's handle and the sequence number to the echo request packet that PE 1 sent out. If the values match, the CLI displays an exclamation point (!).

The following sample output represents what you might see when you issue the **ping mpls ip** and **ping mpls ip detail** commands for the topology shown in Figure 61 on page 292.

```
host1:pe1#ping mpls ip 10.2.2.2/32
Sending 5 UDP echo requests for LDP IPv4 prefix, timeout = 2 sec
!!!!!
Success rate = 100% (5/5), round-trip min/avg/max = 4294967295/4/0 ms

host1:pe1#ping mpls ip 10.2.2.2/32 detail
Sending 5 UDP echo requests for LDP IPv4 prefix, timeout = 2 sec
  MplsNextHopIndex 32 handle 8073311
    '!' - success, 'Q' - request not transmitted,
    '.' - timeout, 'U' - unreachable,
    'R' - downstream router but not destination
    'M' - malformed request, 'N' - downstream router has no mapping
```

```
Sending MPLS ping echo request, handle 8073311 seq 21241
! 10.2.2.2 Replying router is an egress for the FEC at stack depth/0 seq
21241
Sending MPLS ping echo request, handle 8073311 seq 21242
! 10.2.2.2 Replying router is an egress for the FEC at stack depth/0 seq
21242
Sending MPLS ping echo request, handle 8073311 seq 21243
! 10.2.2.2 Replying router is an egress for the FEC at stack depth/0 seq
21243
Sending MPLS ping echo request, handle 8073311 seq 21244
! 10.2.2.2 Replying router is an egress for the FEC at stack depth/0 seq
21244
Sending MPLS ping echo request, handle 8073311 seq 21245
! 10.2.2.2 Replying router is an egress for the FEC at stack depth/0 seq
21245

Success rate = 100% (5/5), round-trip min/avg/max = 4/4/0 ms
```

Packet Flow Example for the trace mpls Command

The following example illustrates the packet flow that results when you issue the **trace mpls ip** command from router PE 1 (10.1.1.1) to router PE 2 (10.2.2.2) over an LDP base tunnel.

host1:pe1#**trace mpls ip 10.2.2.2/32**

1. PE 1 sends an MPLS echo request UDP packet that contains an LDP IPv4 sub-TLV and a Downstream Mapping TLV. The packet has the following attributes:

Source address	10.1.1.1
Destination address	127.0.0.0/8
UDP port	3503
TTL	1
IPv4 prefix in the TLV	10.2.2.2/32
Sender's handle	Randomly generated 32-bit number used to match the reply
Sequence number	Integer that is incremented for each echo request packet

2. The TTL expires on router P 1. P 1 exceptions the packet up to the control plane. Router P 1 then creates an MPLS echo reply packet in reply to the received MPLS echo request. The MPLS echo reply packet has a return code of 8, which means that the packet would have been label-switched at the outermost label (label-stack depth 1). The Downstream Mapping TLV is set to indicate the path that the packet would have taken from the router. The Interface and Label Stack TLV is included in the echo reply packet. The MPLS echo reply packet is sent back as a labeled UDP packet with the following attributes:

Source address	10.3.3.3
Destination address	10.1.1.1
UDP port	3503

3. When the MPLS echo reply reaches router PE 1, the router matches the sender's handle and the sequence number to the echo request packet that PE 1 sent. The CLI displays the router ID of the router that sent the echo reply. The **detail** version of the command displays the downstream mapping TLV contained in the MPLS echo reply.

4. Steps 1–3 are repeated with a TTL of 2 and the destination address set to router P 2's router ID, 10.4.4.4.

5. Router PE 1 next sends an MPLS echo request with a TTL of 3. This packet's TTL expires on router PE 2. PE 2 exceptions the packet up to the control plane. The MPLS trace application on the control plane then creates an MPLS echo reply to the received echo request. The echo reply packet has a return code of 3, which means that the replying router is an egress for the FEC at stack depth. The echo reply packet includes the Interface and Label Stack TLV to indicate both the interface on which the request packet was received and the incoming label stack. The Downstream Mapping TLV is not included in the echo reply packet.

6. When PE 2's echo reply packet reaches router PE 1, the router matches PE 2's handle and the sequence number to the echo request packet that PE 1 sent. The CLI displays the router ID for PE 2, indicating that PE 2 is the target router.

The following sample output represents what you might see when you issue the **trace mpls ip** command for the topology shown in Figure 61 on page 292.

```
host1:pe2#trace mpls ip 10.1.1.1/32
Tracing LDP IPv4 prefix, timeout = 2 sec, Max TTL 32
  MplsNextHopIndex 60, handle 8073312

1 2ms 10.44.44.44 Label switched at stack-depth/1
2 1ms 10.33.33.33 Label switched at stack-depth/1
3 2ms 10.1.1.1 Replying router is an egress for the FEC at stack depth/0
```

Packet Flows for ping and trace to L3VPN IPv4 Prefixes

This example describes packet flow for an MPLS ping is sent from VRF PE 11 on router PE 1 to the IPv4 prefix 10.99.99.21/32. For validation at the remote end, the source address of the echo request packet must be the same as the update-source address of BGP peer.

host1:pe1#**ping mpls l3vpn vrf pe11 10.99.99.21/32**

1. An MPLS echo request packet containing a single VPN IPv4 sub-TLV is sent from PE 1 with the following attributes:

Source address	10.1.1.1
Destination address	127.0.0.0/8
UDP port	3503
TTL	255
Sender's handle	Randomly generated 32-bit number used to match the reply
Sequence number	Integer that is incremented for each echo request packet

The VPN IPv4 sub-TLV has the route distinguisher set to that of the VRF and the IPv4 prefix set to 10.99.99.21/32. The packet exits PE 1 with two labels.

2. Router P 1 switches labels based on the outer label of the packet and forwards the packet to P 2.

3. Router P 2 switches labels based on the outer label of the packet and forwards the packet to PE 2.

4. Router PE 2 pops both labels and determines that the destination address is in the 127.0.0.0/8 subnet. PE 2 sends the packet up to the control plane. The MPLS ping application on the control plane then creates an MPLS echo reply to the received echo request. The echo reply packet has a return code of 3, which means that the replying router is an egress for the FEC at stack depth. The echo reply packet includes the Interface and Label Stack TLV to indicate both the interface on which the request packet was received and the incoming label stack. The MPLS echo reply packet is sent back as a (labeled) UDP packet with the following attributes:

Source address	10.2.2.2
Destination address	10.1.1.1
UDP port	3503

5. When the MPLS echo reply reaches router PE 1, the router matches the sender's handle and the sequence number to the echo request packet that PE 1 sent. The CLI displays an exclamation point (!).

Packet flow for an MPLS trace to an L3VPN IPv4 prefix is the same as for an IPv4 prefix except that the echo request packets and echo reply packets contain the VPN IPv4 sub-TLV instead of the LDP IPv4 sub-TLV. The following sample output represents what you might see when you issue the **trace mpls l3vpn** and **trace mpls l3vpn vrf** commands for the topology shown in Figure 61 on page 292.

```
host1:pe1:pe11#ip8:pe1#trace mpls l3vpn 10.99.99.21/32 detail
Tracing VPN IPv4 prefix, timeout = 2 sec, Max TTL 32
  MplsNextHopIndex 73 handle 8073322

1 0ms 10.33.33.33 Label switched at stack-depth/2
  TLV Interface and Label stack 20 bytes
    Router 10.33.33.33 Intf 10.10.10.2
    [L34 EXP 0 TTL 1] [L68 EXP 0 S TTL 1]
  TLV Downstream mapping 24 bytes
    Router 10.31.31.2 Intf 10.31.31.1 mtu 9180
    [L56 EXP 0 LDP] [L68 EXP 0 S Unknown]
  TLV Downstream mapping 24 bytes
    Router 10.34.34.2 Intf 10.34.34.1 mtu 1500
    [L79 EXP 0 LDP] [L68 EXP 0 S Unknown]
2 2ms 10.55.55.55 Label switched at stack-depth/2
  TLV Interface and Label stack 20 bytes
    Router 10.55.55.55 Intf 10.34.34.2
    [L79 EXP 0 TTL 1] [L68 EXP 0 S TTL 2]
  TLV Downstream mapping 24 bytes
    Router 10.120.120.2 Intf 10.120.120.1 mtu 1500
    [L43 EXP 0 LDP] [L68 EXP 0 S Unknown]
```

```
3 3ms 10.2.2.2 Replying router is an egress for the FEC at stack depth
  TLV Pad 20 bytes
  TLV Interface and Label stack 20 bytes
    Router 10.2.2.2 Intf 10.120.120.2
    [L43 EXP 0 TTL 1] [L68 EXP 0 S TTL 3]

host1:pe1#trace mpls l3vpn vrf pe11 10.99.98.21/32 reply pad-tlv exp-bits 5
detail
Tracing VPN IPv4 prefix, timeout = 2 sec, Max TTL 32
  Handle 1921136  MplsNextHopIndex 78 [L68,L34]

1 0ms 10.33.33.33 Label switched at stack-depth/2
  TLV Pad 20 bytes
  TLV Interface and Label stack 20 bytes
    Router 10.33.33.33 Intf 10.10.10.2
    [L34 EXP 5 TTL 1] [L68 EXP 0 S TTL 1]
  TLV Downstream mapping 24 bytes
    Router 10.31.31.2 Intf 10.31.31.1 mtu 9180
    [L56 EXP 5 LDP] [L68 EXP 0 S Unknown]
  TLV Downstream mapping 24 bytes
    Router 10.34.34.2 Intf 10.34.34.1 mtu 1500
    [L79 EXP 5 LDP] [L68 EXP 0 S Unknown]
2 2ms 10.55.55.55 Label switched at stack-depth/2
  TLV Pad 20 bytes
  TLV Interface and Label stack 20 bytes
    Router 10.55.55.55 Intf 10.34.34.2
    [L79 EXP 5 TTL 1] [L68 EXP 0 S TTL 2]
  TLV Downstream mapping 24 bytes
    Router 10.120.120.2 Intf 10.120.120.1 mtu 1500
    [L43 EXP 5 LDP] [L68 EXP 0 S Unknown]
3 3ms 10.2.2.2 Replying router is an egress for the FEC at stack depth
  TLV Pad 20 bytes
  TLV Interface and Label stack 20 bytes
    Router 10.2.2.2 Intf 10.120.120.2
    [L43 EXP 5 TTL 1] [L68 EXP 0 S TTL 3]
```

Inter-AS Topology

When an L3VPN ping or trace is transmitted, the TTL value on the inner (VPN) label is set to 1 by default. This value causes the TTL to expire on the egress PE of the L3VPN LSP and an echo reply can be sent back to the source. However, in an inter-AS topology, this behavior might result in premature termination of the ping or trace. You can use the **bottom-label-ttl** keyword to avoid this problem.

Packet Flows to L3VPN IPv6 Prefixes

Packet flow for an MPLS ping and trace to an L3VPN IPv6 prefix is the same as for an IPv4 prefix except that the echo request packets and echo reply packets contain the VPN IPv6 sub-TLV instead of the VPN IPv4 sub-TLV.

Configuring IGPs and MPLS

You can use the **tunnel mpls autoroute announce** command to configure a tunnel to announce its endpoint to IS-IS or OSPF so that the IGP can then use the LSP as a shortcut to a destination based on the LSP's metric.

NOTE: This section discusses IS-IS and OSPF; for information about BGP and MPLS, see *Chapter 3, Configuring BGP-MPLS Applications*.

If no tunnels are registered, the IGP calculates the shortest path to a destination by using the shortest path first (SPF) algorithm. The results are represented by the destination node, next-hop address, and output interface, where the output interface is a physical interface.

If you configure an LSP to be announced to the IGP with a certain metric, the LSP appears as a logical interface directly connected to the LSP endpoint. The IGP can consider the LSP as a potential output interface for the LSP endpoint and for destinations beyond the endpoint. In this case, the SPF computation results are represented by the destination node and the output LSP, effectively using the LSP as a shortcut through the network to the destination.

By default, IS-IS and OSPF always use the MPLS tunnel to reach the tunnel endpoint. Best paths determined by SPF calculations are not considered. You can enable the consideration of best paths by issuing the IS-IS or OSPF **mpls spf-use-any-best-path** command. This command causes the IGP to evaluate the LSP as it does any other path. The IGP then either forwards traffic along the best path (which might be the MPLS tunnel), or load-balances between the MPLS tunnel and another path.

The default behavior applies only to reaching the tunnel endpoint itself. For prefixes downstream of the tunnel endpoint, the value of the tunnel metric always determines whether the IGP uses the LSP or the native path, or load-balances between the native path and one or more LSPs.

The tunnel metric can be absolute or relative. An *absolute* metric indicates there is no relationship to the underlying IGP cost. A *relative* metric is added to or subtracted from the underlying IGP shortest path cost.

Example 1 The following commands announce the tunnel to OSPF and specify a relative metric of -2:

```
host1(config-if)#tunnel mpls autoroute announce ospf
host1(config-if)#tunnel mpls autoroute metric relative -2
```

By default, the LSP is preferred to reach the tunnel endpoint. OSPF will treat this LSP as having a metric of 2 less than the shortest path metric it has calculated. The LSP is therefore also preferred over other paths to prefixes beyond the tunnel endpoint.

Example 2 The following commands announce the tunnel to OSPF, specify an absolute metric of 25, and configure OSPF to enable the consideration of SPF best paths:

```
host1(config-if)#tunnel mpls autoroute announce ospf
host1(config-if)#tunnel mpls autoroute metric absolute 25
...
host1(config)#router ospf 1
host1(config-router)#mpls spf-use-any-best-path
```

OSPF uses this metric in its SPF calculations for traffic to the tunnel endpoint as well as beyond the endpoint. Traffic is routed through this LSP only when the other calculated paths have higher metrics.

Configuring the IGPs for Traffic Engineering

For both IGPs, you must issue two commands to enable the IGP to support traffic engineering. See *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 6, Configuring IS-IS* and *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 5, Configuring OSPF* for more information about using these commands.

- IS-IS—Enable the flooding of MPLS traffic-engineering link information into the specified IS-IS level with the **mpls traffic-eng** command. You must also specify a stable router interface with the **mpls traffic-eng router-id** command.

  MPLS traffic engineering also requires that IS-IS generate the new-style TLVs that enable wider metrics. Use the **metric-style wide** command to generate the new-style TLVs. If you are using some IS-IS routers that still cannot interpret the new-style TLVs, use the **metric-style transition** command.

- OSPF—Enable OSPF areas for traffic engineering with the **mpls traffic-eng area** command. OSPF generates opaque LSAs—also known as type-10 opaque link area link states—to flood the traffic-engineering information to the specified area. OSPF builds a traffic-engineering database that it uses in the calculation of shortest path to destinations that satisfy specified traffic-engineering constraints. As with IS-IS, you must also specify a stable router interface with the **mpls traffic-eng router-id** command.

  To enable a multicast network and MPLS traffic engineering (TE) network to interoperate on a router running OSPF, use the **mpls traffic-eng multicast-intact** command.

When you configure a node as the downstream endpoint of an LSP, you must provide a stable interface as the router ID for the endpoint. Typically you select a loopback interface because of its inherent stability. Use the **mpls traffic-eng router-id** command to designate the router as traffic engineering capable and to specify the router ID. For all tunnels that end at this node, set the tunnel destination to the destination node's traffic-engineering router identifier, because the traffic-engineering topology database at the tunnel ingress uses that for its path calculation.

Monitoring Traffic Engineering

The following **show** commands display information about IS-IS traffic engineering:

- **show isis mpls tunnel**—Displays information about any tunnels that are used by IS-IS when calculating next hops. These are tunnels that are either registered with IS-IS when the MPLS tunnel is established or that are explicitly configured as static routes.
- **show isis database verbose**—Displays MPLS traffic-engineering information about the IS-IS database.
- **show isis mpls advertisements**—Displays the last record flooded from MPLS
- **show isis mpls adjacency-log**—Displays a log of the last 20 IS-IS adjacency changes

For OSPF, you can use the **show ip ospf database opaque-area** command to display information about traffic-engineering opaque LSAs.

MPLS and Differentiated Services

Before you read this section, we recommend you be thoroughly familiar with the concepts of the JUNOSe QoS application. For detailed information about QoS, see the *JUNOSe Quality of Service Configuration Guide*.

MPLS employs several strategies to manage different kinds of data streams based on service plans and priority:

- Different conceptual models of diff-serv tunneling that either conceal intermediate LSP nodes from diff-serv operations or render the MPLS network transparent to the diff-serv operations
- Different strategies to set the EXP bits in the shim header to modify or maintain the traffic class/color combination of traffic
- Mapping of traffic behavior aggregates to corresponding per-hop behaviors so that traffic can be differentially switched to the appropriate LSPs to meet your network objectives

Tunneling Models for Differentiated Services

The JUNOSe software supports both the pipe model and the uniform model for tunneling with the **mpls tunnel-model** command. The router also provides a way to implement the functionality of the short pipe model for IP packets.

Pipe and Short Pipe Models

In the pipe and short pipe models, any traffic conditioning (that is, in a pure JUNOSe environment, a change in traffic class/color combination) that is applied when traffic goes through the tunnel has no effect on the EXP bits coding in the inner header. In other words, when traffic exits an LSP (when a label is popped) or when traffic enters an LSP, the inner header's EXP bits coding is not changed.

The pipe and short pipe models differ in the header that the tunnel egress uses when it determines the PHB of an incoming packet. With the short pipe model, the tunnel egress uses an inner header that is used for forwarding. With the pipe model, the outermost label is always used. Because of this, you cannot use PHP with the pipe model.

The pipe model is the default JUNOSe behavior, which you can configure with the **mpls tunnel-model** command. You cannot configure the short pipe model with this command. In fact, on ingress line modules the traffic class/color combination is always determined from the outermost label, so fabric queuing is also based on the outermost label. However, on the egress line module you can achieve the queuing behavior expected with the short pipe model by attaching IP policies to egress interfaces to reset the traffic class/color combinations based on the IP header. However, this method requires that the outgoing packets to be IP. If the outgoing packets are MPLS, then this short pipe model of queuing is not supported.

Uniform Model

The uniform model of tunneling renders MPLS transparent to the differentiated services operation. From the diff-serv perspective, it is as if MPLS is not used. In the uniform model, if traffic conditioning is applied somewhere along the LSP, the EXP bits of the inner header must be changed at the egress when the inner header becomes the outer header (because of the pop of the outer label).

mpls tunnel-model

- Use to specify whether MPLS uses the pipe or uniform model of tunneling for differentiated services.
- Specify the **uniform** keyword for the uniform model and the **pipe** keyword for the pipe model.
- Example

  host1(config)#**mpls tunnel-model uniform**

- Use the **no** version to restore the default, the pipe model.

EXP Bits and Differentiated Services

MPLS matches on the EXP bits for incoming traffic to set the traffic class/color combination, and sets the EXP bits for outgoing traffic based on the traffic class/color combination.

Incoming Traffic

For incoming MPLS traffic, the traffic class/color combination is set according to the EXP bits in the outermost label, either per the policy attached to the label or per the per-VR rules. The policy has precedence over the per-VR rules. Therefore, fabric queuing is always based on the outer label's EXP bits.

If the traffic is label-switched through the router, the EXP bits value associated with the incoming label that is used for switching—which can be either an outermost label or an inner label after popping one or more outer labels—is passed onto the egress line module. This behavior enables the EXP bits value to be copied to outgoing labels, used to reset the traffic class/color combination on the egress module, or both.

Outgoing Traffic

Outgoing traffic is queued according to traffic class/color combinations. The applied combination can be the same as was set on the ingress line module, or it can be reset on the egress line module by egress IP policy.

Figure 62 on page 303 illustrates how the initial value of the EXP bits is set for the first label pushed. Figure 63 on page 304 illustrates how the EXP bits can be changed for all labels, including the first label, by attached policies or per-VR EXP rules. The following section describes in detail how the EXP bits value is set for outgoing traffic.

Setting the EXP Bits for Outgoing Traffic

Different types of packets distributed into LSPs by the router have different default settings for the EXP bits. For IP packets, the EXP bits value is set to match the IP precedence value from the TOS field of the packet header. For non-IP packets, such as Martini or VPLS packets, the EXP bits value is set to 000. You can use the **mpls copy-upc-to-exp** command to free the EXP bits value in IP packets from being tied to the IP precedence value. Instead, this command sets the EXP bits value to match the user packet class (UPC) value.

The IP precedence value can be copied back into the IP precedence field of the IP packet header at the LSP endpoint on the ingress line module. This action takes place only if the IP header is exposed after popping the MPLS labels and if the uniform tunnel model is employed. The remaining bits of the TOS field are not touched.

In contrast, when you issue **mpls copy-upc-to-exp** command, the EXP bits value is not copied to the UPC field at the LSP endpoint, because the UPC value might have been set by a lower layer policy for a different purpose.

NOTE: For control traffic originated from this router, if an attached per-LSP policy has rules to modify the EXP bits, or if per-VR EXP rules are configured, the EXP bits value copied from the IP precedence value might be overwritten incorrectly because the default traffic class/color combination for control traffic is best-effort/green. You can avoid this situation by establishing an outgoing IP policy that sets the traffic class/color combination for control traffic so that the policy or rules have the correct traffic class/color to work with.

If per-LSP policies are used or per-VR rules are configured, by default all labels pushed by the router for the same packet have the same EXP bits value. That value is determined by the policies or rules.

You can use the **mpls preserve-vpn-exp** command to specify that the EXP bits value for the VPN or Martini or VPLS label pushed by the router cannot be modified by either policy for outer labels or by per-VR rules. This capability is useful if you want the inner labels to have a different value for the EXP bits than do the outer labels. For example, in a VPN you might want the inner label's EXP bits value to be the copied IP precedence value. You might want the base label's EXP bits value set according to the mapping of EXP bits to traffic class/color combination that is defined in your network.

Figure 62: Flow for Initial Setting of EXP Bits for the First Label Pushed

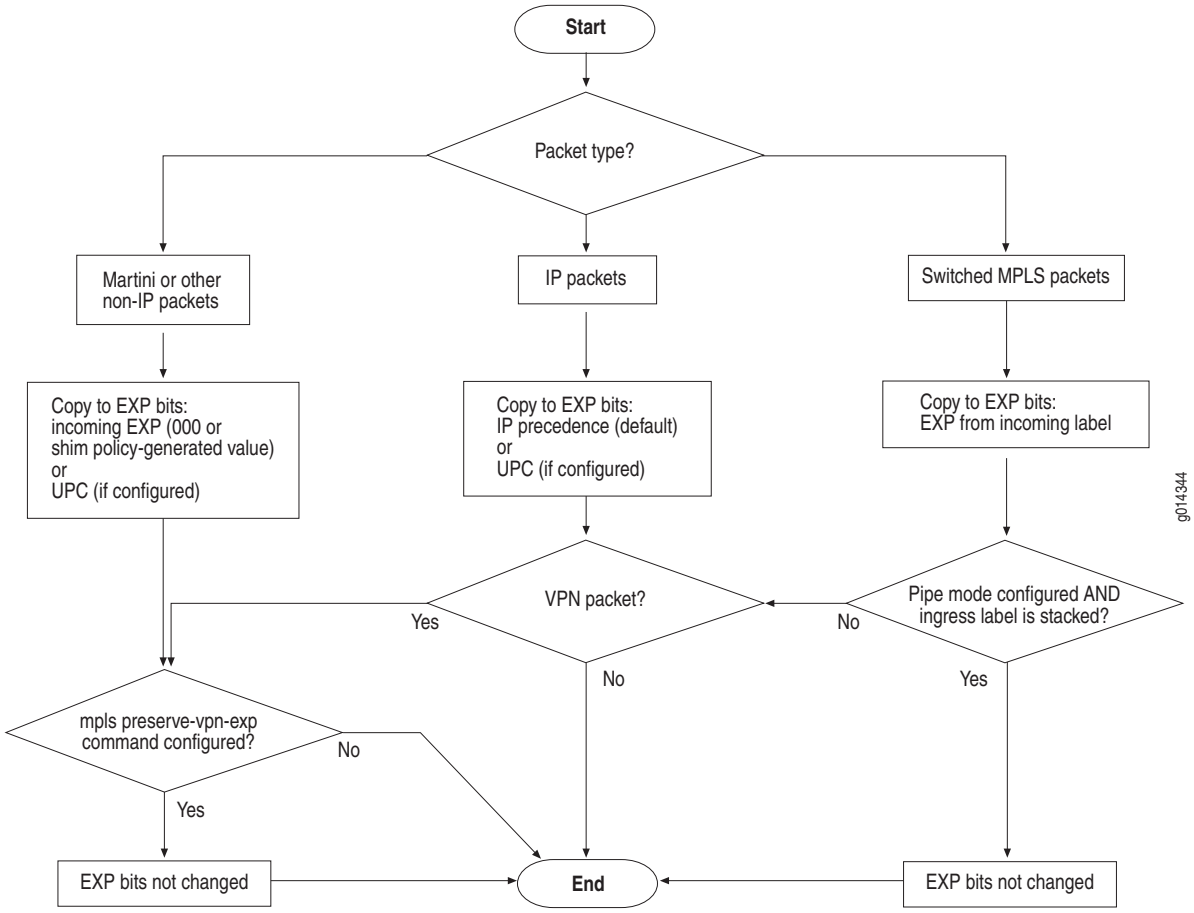
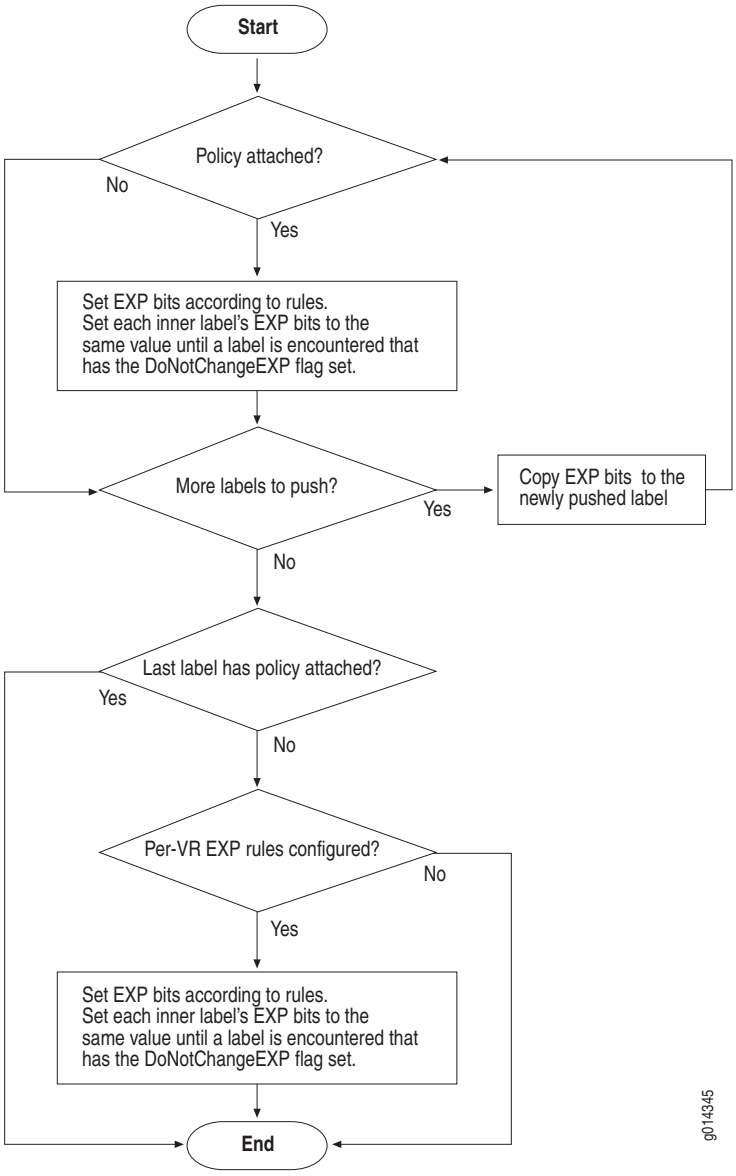


Figure 62 shows how packet type and configuration determine how the EXP bits are set for the first label pushed.

Figure 63: Flow for Setting EXP Bits for All Pushed Labels



mpls copy-upc-to-exp

- Use to set the initial value of the EXP bits to the UPC value associated with the packets
- For IP packets, the default value of the EXP bits is set to the value of the IP precedence field. For non-IP packets, the default value of the EXP bits is set to 000.
- Example

  host1(config)#**mpls copy-upc-to-exp**

- Use the **no** version to restore the default condition.

mpls preserve-vpn-exp

- Use to prevent the value of the EXP bits for a VPN/VC label from being modified by a per-LSP policy applied for the outer labels or by per-VR traffic class/color rules.
- By default, per-LSP policies or per-VR rules modify all labels in a given label stack to have the same value for the EXP bits.
- Example

  host1(config)#**mpls preserve-vpn-exp**

- Use the **no** version to restore the default condition.

Example Differentiated Services Application

Figure 64 shows an example topology where a service provider offers the following differentiated services to its customers over its MPLS network:

- QoS Internet service—The CE router is managed by the provider and sets the IP precedence to predefined values. IP policy on the PE router sets the traffic-class/color combination according to the incoming well-defined IP precedence value. The policy also sets the UPC value to the incoming well-defined IP precedence value.
- Plain Internet service—IP policy on the PE router leaves the traffic-class/color combination as the default value, best-effort/green. The policy sets the UPC to 0.
- QoS VPN service—For CE-to-PE traffic, the VPN EXP is copied from the IP precedence value when the PE router pushes VPN stacked labels.

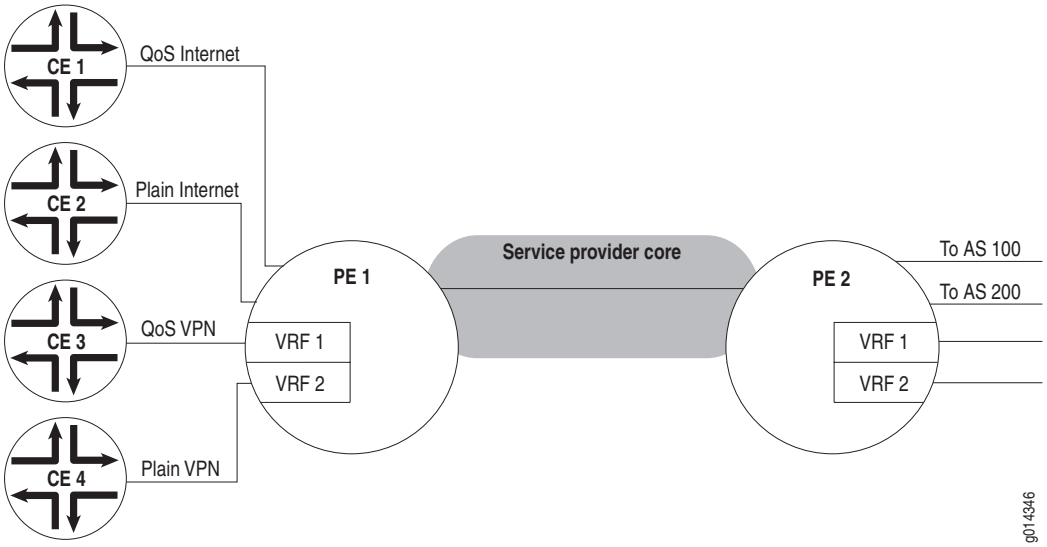
  For PE-to-CE traffic, IP policy on the PE router resets the traffic-class/color combination according to the received, well-defined IP precedence value, so that egress queuing is based on the IP precedence value. This action takes place on the egress line module.
- Plain VPN service—For CE-to-PE traffic, the VPN EXP bits are set to 000 when the PE router pushes VPN stacked labels.

  For PE-to-CE traffic, IP policy on the PE router resets the traffic-class/color combination to the default value, best-effort/green, so that packets are queued as best-effort. The IP precedence value is left unchanged.

In this example, the provider also offers an inter-AS VPN service. The provider's own protocol traffic, for example, BGP signaling traffic such as update messages, is also labeled, with the EXP bits set to the same value as the IP precedence.

The egress queuing of traffic as it leaves the provider is always based on either the VPN EXP bits as received on the core side in inter-AS case, or the IP precedence value in all other cases. It is acceptable that fabric queuing is based on the incoming base label's EXP.

Figure 64: Differentiated Services over an MPLS Network



Configuration Example

To configure the differentiated services described in this example:

1. Create and attach an IP input policy for the QoS Internet service to CE interfaces on the PE router for incoming traffic.

   ```
   host1(config)#ip classifier-list prec0 ip any any precedence 0
   host1(config)#ip classifier-list prec1 ip any any precedence 1
   …
   host1(config)#ip policy-list qos-service
   host1(config-policy-list)#classifier-group prec0
   host1(config-policy-list-classifier-group)#user-packet-class 0
   host1(config-policy-list-classifier-group)#traffic-class class0
   host1(config-policy-list-classifier-group)#color green
   host1(config-policy-list)#classifier-group prec1
   host1(config-policy-list-classifier-group)#user-packet-class 1
   host1(config-policy-list-classifier-group)#traffic-class class1
   host1(config-policy-list-classifier-group)#color green
   …
   host1(config)#interface atm 3/0.1
   host1(config-subif)#ip policy input qos-service
   ```

2. Create and attach an IP input policy for the plain Internet service to CE interfaces on the PE router for incoming traffic. All traffic is treated as best effort, so no classifier group is necessary.

   ```
   host1(config)#ip policy-list plain-service
   host1(config-policy-list-classifier-group)#user-packet-class 0
   host1(config-policy-list-classifier-group)#traffic-class best-effort
   host1(config-policy-list-classifier-group)#color green

   host1(config)#interface atm 5/0.1
   host1(config-subif)#ip policy input plain-service
   ```

3. Attach an IP output policy for the QoS VPN service to CE interfaces on the PE router for outgoing traffic. The same qos-service policy that is attached to the input in Step 1 can be used on the output, even though the UPC setting is not needed.

   host1(config)#**Interface atm 3/0.1**
   host1(config-subif)#**Ip policy output qos-service**

   Attach an IP output policy for the plain VPN service to CE interfaces on the PE router for outgoing traffic. The same plain-service policy that is attached to the input in Step 2 can be used on output, although the UPC setting is not needed.

   host1(config)#**Interface atm 5/0.1**
   host1(config-subif)#**Ip policy output plain-service**

4. For traffic toward the core, configure per-VR rules or per-LSP policies to set the base EXP bits value according to the traffic-class/color combination. Issue the **mpls copy-upc-to-exp** command to set the VPN EXP bits value to the UPC value. The UPC value is the same as the IP precedence value for the QoS service case; for all other cases the value is 000. Configure the **mpls preserve-vpn-exp** command so that VPN EXP bits are not subject to policy or to per-VR EXP rules.

   host1(config)#**mpls match traffic-class … color … set exp-bits …**
   host1(config)#**mpls copy-upc-to-exp**
   host1(config)#**mpls preserve-vpn-exp**

   You must attach a policy to the core-side IP interface to set the UPC value of the control traffic appropriately so that the EXP bits value is copied from the UPC when this traffic goes out as MPLS packets.

   host1(config)#**ip classier-list control-traffic-prec0 …**
   host1(config)#**ip classier-list control-traffic-prec1 …**
   …
   host1(config)#**ip policy-list core-ip-policy**
   host1(config-policy-list)#**classifier-group control-traffic-prec0**
   host1(config-policy-list-classifier-group)#**user-packet-class prec0**
   host1(config-policy-list-classifier-group)#**traffic-class class0**
   host1(config-policy-list-classifier-group)#**color green**
   host1(config-policy-list)#**classifier-group control-traffic-prec1**
   host1(config-policy-list-classifier-group)#**-packet-class prec1**
   host1(config-policy-list-classifier-group)#**traffic-class class1**
   host1(config-policy-list-classifier-group)#**color green**
   …

   host1(config)#**interface pos 0/0**
   host1(config-subif)**ip policy output core-ip-policy**

5. For traffic from the core, configure per-VR rules or per-LSP policies to set the traffic-class/color combination—and therefore shape the egress traffic queue—according to the value of the EXP bits in the base label. This action causes

   host1(config)#**mpls match exp-bits** <value> **set traffic-class <**className**> color**
   …

Classifying Traffic for Differentiated Services

In a differentiated services domain, traffic is classified into a behavior aggregate (BA), based on the type of diff-serv behavior for the traffic. At each node, traffic belonging to a particular BA is mapped to the corresponding per-hop behavior (PHB), which provides the scheduling behavior and drop probability required by the traffic.

MPLS uses the EXP bits in the shim header to support differentiated services. The JUNOSe software supports both statically configured and signaled mapping between the EXP bits and the PHB of traffic.

In a signaled environment, you can configure on the ingress node the set of PHBs that a tunnel supports, and then the set of PHBs is signaled end to end.

To support differentiated services, MPLS employs two types of LSPs: E-LSPs and L-LSPs. The two types differ in how their PHB is determined. In the JUNOSe software, the PHB is a combination of traffic class (also called per-hop scheduling class, or PSC) and drop precedence (color).

- E-LSPs (EXP-inferred-PSC LSP) can transport as many as eight BAs. For E-LSPs, the traffic's PHB is learned from the MPLS shim header.
- L-LSPs (Label-only-inferred-PSC LSP) transport a single PSC. The PHB is determined from a combination of the packet's label, which indicates the traffic class, and the EXP field of the shim header, which indicates the drop precedence.
    - Table 29 indicates how the PSC (column 1) is combined with the EXP field (column 2) to determine the PHB for incoming traffic on L-LSPs.

Table 29: Incoming L-LSP PHB Determination

PSC	+	EXP Field	=	PHB
BE		000		BE
CSn		000		CSn
AFn		001		AFn1
AFn		010		AFn2
AFn		011		AFn3
EF		000		EF

For nonstandard PHBs (any that are not listed in Table 29), the JUNOSe software uses mapping similar to AFn mapping; EXP 001 is mapped to color green, EXP 010 is mapped to yellow, and EXP 011 is mapped to red.

Table 30 presents three examples that indicate how the PSC and the EXP field are combined to determine the PHB for traffic on incoming L-LSPs.

Table 30: Examples of Incoming L-LSP PHB Determination

PSC	+	EXP Field	=	PHB
AF2		010		AF22
AF3		010		AF32
AF3		011		AF33

- For outgoing L-LSPs, the EXP is determined by the PHB. Table 31 indicates the PHB-to-EXP mapping for outgoing traffic on L-LSPs.

Table 31: Outgoing L-LSP PHB Determination

PHB	=	EXP Field
BE		000
CSn		000
AFn1		001
AFn2		010
AFn3		011
EF		000

For nonstandard PHBs, the mapping is similar to AFn mapping. Red color maps to 011, yellow maps to 010, and green maps to 001.

Configured Mapping

You can configure static EXP-to-PHB mapping at the per-VR level. Configured mapping applies regardless of label distribution protocol, BGP. LDP, or RSVP-TE.

The PHB of incoming packets is determined from the EXP bits according to the following command:

```
mpls match exp-bits bitValue set traffic-class className color { green |
yellow | red }
```

The EXP bits of outgoing packets are determined from the PHB according to the following command:

```
mpls match traffic-class className color { green |
yellow | red } set exp-bits bitValue
```

The configuration applies only to LSPs that do not have specific policies attached (by either per-LSP configured mapping or signaled mapping).

mpls match exp-bits

- Use to set a combination of traffic class and color for incoming traffic that matches the specified EXP bits value in the shim header.
- Specify an integer value in the range 0–7 to match the corresponding binary values (000–111) for the three EXP bits in the shim header.
- You can repeat the command to support the eight possible EXP bit values.
- Example

  host1(config)#**mpls match exp-bits 1 set traffic-class bronze color red**

- Use the **no** version to restore the default behavior, setting neither traffic class nor color for all traffic matching the specified EXP bits value.

mpls match traffic-class

- Use to set the EXP bits in the shim header of outgoing traffic that matches a particular combination of traffic class and color.
- Specify an integer value in the range 0–7 to set the corresponding binary values (000–111) for the three EXP bits in the shim header.
- You can repeat the command to support up to 24 combinations: eight traffic classes supported on the router times three colors.
- Example

  host1(config)#**mpls match traffic-class gold color green set exp-bits 7**

- Use the **no** version to restore the default behavior for traffic that matches the specified traffic class and color. For matching traffic entering an LSP, it sets the EXP bits to 000. For matching transit traffic, it does nothing.

Signaled Mapping for RSVP-TE Tunnels

For signaled mapping between EXP and PHB, policies apply the EXP bits matching and setting on a per-LSP basis rather than a per-VR basis. Signaled mapping applies only when RSVP-TE is the label distribution protocol.

When traffic is mapped onto the ingress router of the LSP, the EXP bits are set according to a policy attached to the LSP. The policy corresponds to the EXP-to-PHB mapping defined for the LSP. Typically, the policy sets the EXP bits differently according to classifier lists that match on internal class/color information or on a user packet class associated with a packet.

For transit routers and egress routers along the path of the LSP, the incoming EXP bits are matched to determine the traffic class and drop preference (color red, yellow, or green). This matching is accomplished by means of a policy corresponding to the signaled EXP-to-PHB mapping that is created and attached when the LSP is established.

EXP bits are not normally changed on transit routers, but when traffic is sent out of an LSP on a transit router, the bits can be changed by the policy. Normally, however, the net effect is that the EXP-bits remain the same through the mapping sequence of EXP bits to an internal traffic class/color combination back to EXP bits, unless the traffic class/color combination is also modified by other factors.

Because the policy (which maps the EXP bits to an internal traffic class/color combination and vice versa) attached to an LSP is created according to the PHB-ID–to–EXP mapping signaled by RSVP-TE, you must configure on each router a mapping association between PHB IDs and the internal traffic class/color combinations.

The JUNOSe software automatically generates and attaches policies when tunnels are established.

Figure 65 shows the mapping associations between PHB IDs, EXP bits, and traffic class (TC)/color combination in an E-LSP case.

- Mapping association between PHB ID and EXP bits is configured on ingress routers using the **tunnel mpls diff-serv phb-id** command.
- Mapping association between PHB ID and traffic class/color combination is configured on all routers using the **mpls diff-serv phb-id traffic-class** command.
- Mapping association between EXP bits and traffic class/color combination is done automatically by the JUNOSe software at the appropriate routers along the path.

Figure 65: Associations Between PHB ID, EXP Bits, and Traffic Classes/Colors

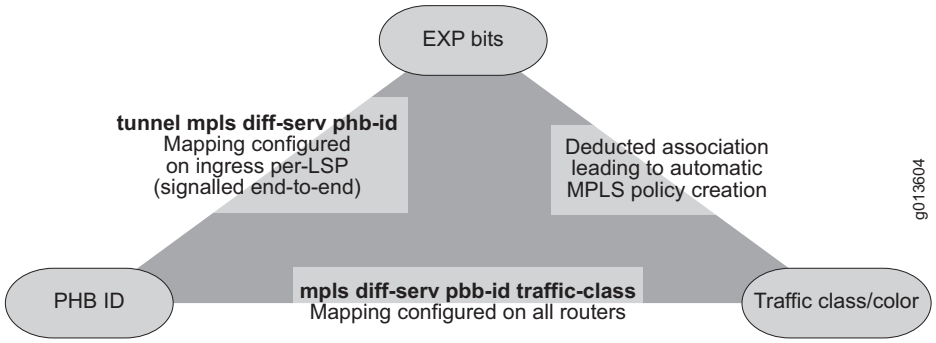
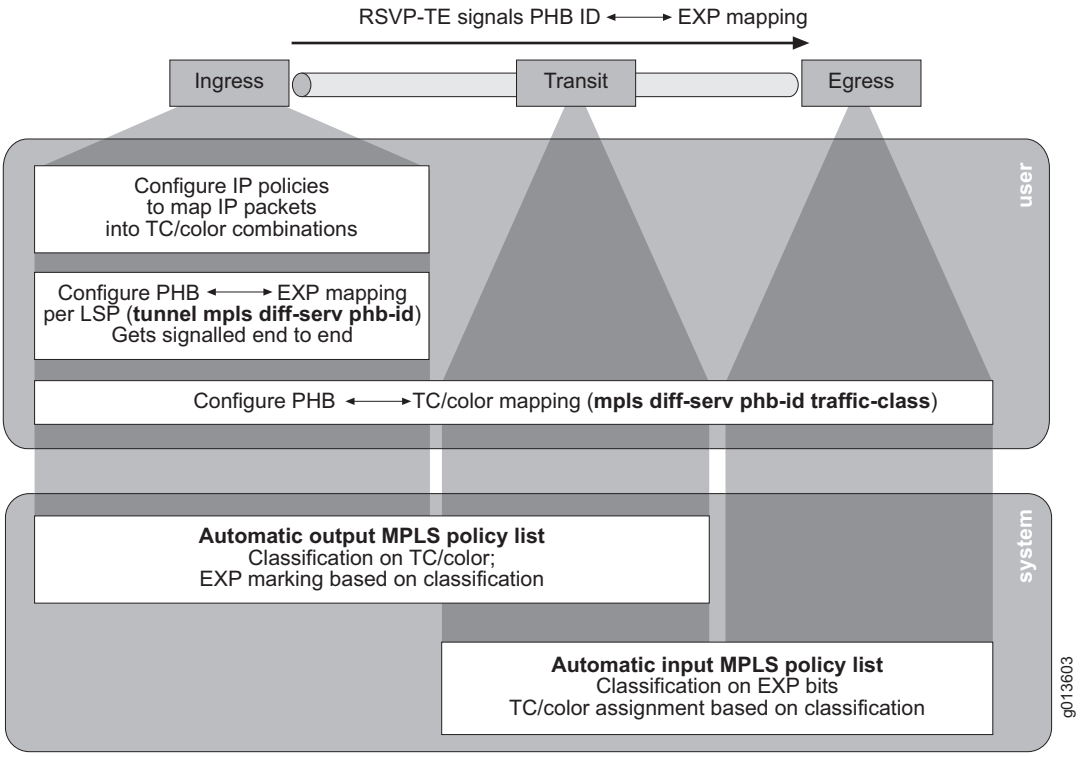


Figure 66 shows the operations performed at ingress, transit, and egress systems during signaled mapping sessions.

Figure 66: Signaled Mapping



mark-exp

- Use to define a policy rule that sets the EXP bits in packets to which the policy is applied.
- Use the **mask** keyword to modify certain EXP bits present in the packet.
- Example

  host1(config-policy-list)#**mark-exp 5 classifier-group claclEXP precedence 32**

- Use the **suspend** version to temporarily suspend the EXP bits policy rule. Use the **no suspend** version to resume the application of the suspended rule. Use the **no** version to remove the rule from the policy list.

mpls classifier-list

- Use to create or modify an MPLS classifier control list to match on traffic class/color combination or EXP bits.
- Example

  host1(config)#**mpls classifier-list be-green traffic-class best-effort color yellow**

- Use the **no** version to delete the classifier control list.

mpls diff-serv phb-id traffic-class

- Use to map the specified PHB ID to the internal traffic class/color combination. If color is specified, the PHB ID can be used only for E-LSPs. If color is *not* specified, the PHB ID can be used only for L-LSPs.
- Example

  host1(config)#**mpls diff-serv phb-id standard 45 traffic-class gold color green**

- Use the **no** version to remove the mapping.

mpls policy-list

- Use to create or modify an MPLS policy.
- This command accesses Policy List Configuration mode, from which you define the rules that make up the MPLS policy. See the *JUNOSe Policy Management Configuration Guide* for more information about defining policies.
- Example

  host1(config)#**mpls policy-list mpls-exp-setting**

- Use the **no** version to delete the policy.

mpls policy-statistics

- Use to enable collection of policy statistics for a tunnel or LSP. Collection is disabled by default.
- Policy statistics are displayed when you issue the **show mpls forwarding** or **show mpls tunnel** command, if a policy is attached and policy statistics are enabled.
- Example

  host1#**mpls policy-statistics boston2dc**

- Use the **disable** version to stop collection of policy statistics. There is no **no** version.

mpls traffic-class

- Use to specify the traffic class for which LSP-level queues are created and the scheduler profile to be used with the queues.
- The scheduler profile distributes bandwidth among different classes.

  Classes for which the LSP-level queues are created originate from one of two sources:

  - E-LSPs and L-LSPs—Classes derived from the signaled PHB-ID
  - Regular LSPs—Classes configured with the **mpls traffic-class** command

- Example

  host1(config)#**mpls traffic-class af1 scheduler-profile af1-scheduler-profile**

- Use the **no** version to remove the configuration.

tunnel mpls diff-serv phb-id

- Use to specify the PHB supported by a signaled tunnel.
- For E-LSPs, you also use this command to map the PHB to the specified exp-bits *bitValue*. You can repeat the command for up to eight PHB mappings.
- Example

  host1(config-if)#**tunnel mpls diff-serv phb-id standard 35 exp-bits 5**

- For L-LSPs, do not use the **exp-bits** keyword. If you repeat the command, the most recent command overwrites the previous command.
- Example

  host1(config-if)#**tunnel mpls diff-serv phb-id private 40**

- Use the **no** version to remove the mapping association.

Preference of per-VR Versus per-LSP Behavior

MPLS always prefers the per-LSP method of matching and setting EXP bits by means of applied policies over the per-VR method.

Per-VR matching of EXP bits is not performed on the LSP when an input policy (matching on incoming EXP bits) is attached to the ingress segment of the LSP.

Similarly, per-VR setting of EXP bits is not performed on the LSP when an output policy (setting the outgoing EXP bits) is attached to the egress segment of the LSP.

Example Configuration

The commands in this example illustrate a partial network configuration that supports four differentiated service classes on a particular tunnel: a best-effort class, two assured forwarding classes, and an expedited forwarding class. Table 32 presents the mapping between EXP bits, PHB, PHB ID, and traffic class/color combination.

Table 32: Differentiated Services Mapping

EXP	PHB	PHB ID	6-bit PHB ID	Traffic Class/Color
000	BE	0x0000	00	best-effort/green
001	AF11	0x2800	10	af1/green
010	AF12	0x3000	12	af1/yellow
011	AF13	0x3800	14	af1/red
100	AF21	0x4800	18	af2/green
101	AF22	0x5000	20	af2/yellow
110	AF23	0x5800	22	af2/red
111	EF	0xb800	46	ef/green

NOTE: This example includes both MPLS and policy configuration commands, and assumes that you are thoroughly familiar with the information and commands presented in the *JUNOSe Policy Management Configuration Guide*.

The four traffic classes are configured to allocate fabric resources and allow global synchronization of the three segments of the data path through an E-series router: ingress, fabric, and egress. The JUNOSe software automatically creates the best-effort traffic class, with a default weight of eight. You must define the remaining three classes, af1, af2, and ef. In this example, the af1 class has twice as much fabric bandwidth as the best-effort class, and the af2 class has twice as much fabric bandwidth as the af1 class. The expedited forwarding traffic (the ef class) requires strict-priority queuing.

```
host1(config)#traffic-class af1
host1(config-traffic-class)#fabric-weight 16
host1(config)#traffic-class af2
host1(config-traffic-class)#fabric-weight 32
host1(config)#traffic-class ef
host1(config-traffic-class)#fabric-strict-priority
```

Define two scheduler profiles for the af1 and af2 classes on the egress line modules:

```
host1(config)#scheduler-profile af1-scheduler-profile
host1(config-scheduler-profile)#weight 16
host1(config)#scheduler-profile af2-scheduler-profile
host1(config-scheduler-profile)#weight 32
```

Create queue profiles to define how queues are instantiated to implement the corresponding traffic classes and PHBs. The JUNOSe software automatically creates the best-effort queue profiles.

```
host1(config)#queue-profile af1-queues
[Queue configuration omitted]
host1(config)#queue-profile af2-queues
[Queue configuration omitted]
host1(config)#queue-profile ef-queues
[Queue configuration omitted]
```

The scheduler and queue profiles are referenced in QoS profiles. For example, you can create a QoS profile for port-based per-class queuing or for LSP-level per-class queuing (configuration omitted).

You must map the PHB IDs to the appropriate traffic class/color combinations:

```
host1(config)#mpls diff-serv phb-id standard 0 traffic-class best-effort color green
host1(config)#mpls diff-serv phb-id standard 10 traffic-class af1 color green
host1(config)#mpls diff-serv phb-id standard 12 traffic-class af1 color yellow
host1(config)#mpls diff-serv phb-id standard 14 traffic-class af1 color red
host1(config)#mpls diff-serv phb-id standard 18 traffic-class af2 color green
host1(config)#mpls diff-serv phb-id standard 20 traffic-class af2 color yellow
host1(config)#mpls diff-serv phb-id standard 22 traffic-class af2 color red
host1(config)#mpls diff-serv phb-id standard 46 traffic-class ef color green
```

Configuration on the Ingress Router

You must access the tunnel interface to map the PHB IDs to the EXP bits. The E-series router signals this mapping to all routers on the tunnel. You can establish different PHB-ID–to–EXP mappings for different tunnels.

```
host1(config)#interface tunnel mpls:example
```

PHB-ID–to–EXP mapping for the best-effort traffic class:

```
host1(config-if)#tunnel mpls diff-serv phb-id standard 0x0000 exp-bits 0
```

PHB-ID–to–EXP mapping for the af1 traffic class:

```
host1(config-if)#tunnel mpls diff-serv phb-id standard 10 exp-bits 1
host1(config-if)#tunnel mpls diff-serv phb-id standard 12 exp-bits 2
host1(config-if)#tunnel mpls diff-serv phb-id standard 14 exp-bits 3
```

PHB-ID–to–EXP mapping for the af2 traffic class:

```
host1(config-if)#tunnel mpls diff-serv phb-id standard 18 exp-bits 4
host1(config-if)#tunnel mpls diff-serv phb-id standard 20 exp-bits 5
host1(config-if)#tunnel mpls diff-serv phb-id standard 22 exp-bits 6
```

PHB-ID–to–EXP mapping for the ef traffic class:

```
host1(config-if)#tunnel mpls diff-serv phb-id standard 46 exp-bits 7
```

Define classifier control lists to classify the incoming packets into classifier groups. Although not shown here, for each CLACL you must define the rules that will select the appropriate incoming packets: be, af1, af2, or ef.

```
host1(config)#classifier-list be-packets
host1(config)#classifier-list af1-packets
host1(config)#classifier-list af2-packets
host1(config)#classifier-list ef-packets
```

Define a policy that maps the selected packets into traffic classes. For the assured forwarding classes, this example uses rate limit profiles to set the colors.

```
host1(config)#policy-list classify-packets
host1(config-policy-list)#traffic-class best-effort classifier-group bf-packets
host1(config-policy-list)#traffic-class ef classifier-group ef-packets
host1(config-policy-list)#traffic-class af1 classifier-group af1-packets
host1(config-policy-list)#traffic-class af2 classifier-group af2-packets
host1(config-policy-list)#rate-limit-profile af1-profile classifier-group af1-packets
host1(config-policy-list)#rate-limit-profile af2-profile classifier-group af2-packets
host1(config)#rate-limit-profile af1-profile
host1(config-rate-limit-profile)#committed-rate 6000000
host1(config-rate-limit-profile)#committed-burst 1000000
host1(config-rate-limit-profile)#peak-rate 8000000
host1(config-rate-limit-profile)#peak-burst 1000000
host1(config)#rate-limit-profile af2-profile
host1(config-rate-limit-profile)#committed-rate 8000000
host1(config-rate-limit-profile)#committed-burst 1500000
host1(config-rate-limit-profile)#peak-rate 12000000
host1(config-rate-limit-profile)#peak-burst 1000000
```

You attach the policy to the ingress interface of the ingress router. As packets arrive, they are classified with the internal traffic class/color combination and forwarded into the appropriate queues in the fabric. When the packets are sent into the tunnel out of the ingress router, the EXP bits are set according to the router-generated policy (in this example called mpls-exp-setting) that the JUNOSe software automatically attached to the tunnel.

Configuration on the Ingress and Transit Routers

When the tunnel is established, the JUNOSe software automatically creates an output policy to map traffic-class/color combinations to EXP bits and attaches the policy to the outgoing segment of the tunnel. The JUNOSe software generates classifier list and policy list names, and creates the EXP-setting policy as if the following commands were entered:

NOTE: You do not actually issue these commands; they represent the behavior automatically performed by the router.

```
host1(config)#mpls classifier-list be-green traffic-class best-effort color green
host1(config)#mpls classifier-list ef-green traffic-class ef color green
host1(config)#mpls classifier-list af1-green traffic-class af1 color green
host1(config)#mpls classifier-list af1-yellow traffic-class af1 color yellow
host1(config)#mpls classifier-list af1-red traffic-class af1 color red
host1(config)#mpls classifier-list af2-green traffic-class af2 color green
host1(config)#mpls classifier-list af2-yellow traffic-class af2 color yellow
host1(config)#mpls classifier-list af2-red traffic-class af2 color red
host1(config)#mpls policy-list mpls-exp-setting
host1(config-policy-list)#mark 0 classifier-group be-green
host1(config-policy-list)#mark 1 classifier-group af1-green
host1(config-policy-list)#mark 2 classifier-group af1-yellow
host1(config-policy-list)#mark 3 classifier-group af1-red
host1(config-policy-list)#mark 4 classifier-group af2-green
host1(config-policy-list)#mark 5 classifier-group af2-yellow
host1(config-policy-list)#mark 6 classifier-group af2-red
host1(config-policy-list)#mark 7 classifier-group ef-green
```

NOTE: For a topology-driven LSP, you have to configure and apply the classifier list and policy list manually.

Configuration on the Transit and Egress Routers

When the tunnel is established, the JUNOSe software automatically creates an input policy to match the EXP bits and map them to the traffic-class/color combinations and attaches the policy to the incoming segment of the tunnel. The JUNOSe software generates classifier list and policy list names, and creates the policy as if the following commands were entered:

NOTE: You do not actually issue these commands; they represent the behavior automatically performed by the router.

```
host1(config)#mpls classifier-list bf-packets exp 0
host1(config)#mpls classifier-list af11-packets exp 1
```

```
host1(config)#mpls classifier-list af12-packets exp 2
host1(config)#mpls classifier-list af13-packets exp 3
host1(config)#mpls classifier-list af21-packets exp 4
host1(config)#mpls classifier-list af22-packets exp 5
host1(config)#mpls classifier-list af22-packets exp 6
host1(config)#mpls classifier-list ef-packets exp 7
host1(config)#mpls policy-list mpls-exp-matching
host1(config-policy-list)#traffic-class best-effort classifier-group bf-packets
host1(config-policy-list)#traffic-class af1 classifier-group af11-packets
host1(config-policy-list)#traffic-class af1 classifier-group af12-packets
host1(config-policy-list)#traffic-class af1 classifier-group af13-packets
host1(config-policy-list)#traffic-class af2 classifier-group af21-packets
host1(config-policy-list)#traffic-class af2 classifier-group af22-packets
host1(config-policy-list)#traffic-class af2 classifier-group af23-packets
host1(config-policy-list)#traffic-class ef classifier-group ef-packets
host1(config-policy-list)#color green classifier-group af11-packets
host1(config-policy-list)#color green classifier-group af21-packets
host1(config-policy-list)#color yellow classifier-group af12-packets
host1(config-policy-list)#color yellow classifier-group af22-packets
host1(config-policy-list)#color red classifier-group af13-packets
host1(config-policy-list)#color red classifier-group af23-packets
```

NOTE: For a topology-driven LSP, you must configure and apply the classifier list and policy list manually.

The packets are forwarded to the appropriate fabric queue according to the traffic class/color combination. On a transit router, when the packet is forwarded out of the tunnel, the router-generated output policy then sets the EXP bits back according to the traffic class/color combination. Typically, the effect of the EXP bits to traffic class/color combination to EXP bits is no change.

On an egress router, where the tunnel terminates, no router-generated output policy is attached, and the packets pass out of the router subject to any manually configured IP policy management applied to their traffic class/color combination.

Monitoring MPLS

Use the **show** commands in this section to monitor MPLS status.

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot*/*adapter*/*port*).

You can set a statistics baseline for MPLS major interface statistics with the **baseline mpls interface** command. The following statistics are maintained for each MPLS major interface:

- receive packets and octets
- receive discarded packets
- receive error packets
- transmit packets and octets
- transmit discarded packets
- transmit error packets

- failed label lookups

You can set a statistics baseline for MPLS forwarding table entries with the **baseline mpls label** command. You must first enable the statistics with the **mpls statistics label** command. When enabled, the following statistics are maintained for each forwarding table entry:

- receive packets and octets
- receive discarded packets
- receive error packets

You can set a statistics baseline for MPLS next-hop table entries with the **baseline mpls next-hop** command. You must first enable these statistics with the **mpls statistics next-hop** command. When enabled, the following statistics are maintained for each next-hop table entry:

- out packets and bytes
- out discarded packets
- out error packets

You can set a statistics baseline for MPLS tunnel statistics with the **baseline mpls tunnel** command.

You can enable collection of the following statistics for each policy attached to a tunnel by issuing the **mpls statistics policy** command:

- packets and bytes
- classifier group
- EXP bits value

You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. See *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface* for details.

You can trace paths through the MPLS user plane with the **traceroute** command. ICMP extensions enable LSRs to append MPLS header information (the label stack) to ICMP destination unreachable and time exceeded messages. The **traceroute** display shows the label and EXP bits used to switch the ICMP packets:

```
host1:edge1#traceroute 10.90.101.9
Tracing route to 10.90.101.9, TTL = 32, timeout = 2 sec.
(Press ^c to stop.)
1  3ms  2ms  2ms       10.90.101.4    mplsLabel1=4009
mplsExpBits1=0
2  2ms  2ms  2ms       10.90.101.7    mplsLabel1=7004
mplsExpBits1=0
3  2ms  2ms  2ms       10.90.101.9
```

See *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 1, Configuring IP*, for more information about using the **traceroute** command.

baseline mpls interface

- Use to set the baseline on all MPLS major interface statistics to zero on the specified interface.
- Example

  host1#**baseline mpls interface**

- There is no **no** version.

baseline mpls label

- Use to set the baseline on all MPLS forwarding table statistics to zero on the specified interface or for the specified label.
- Example

  host1#**baseline mpls label**

- There is no **no** version.

baseline mpls next-hop

- Use to set a statistics baseline for the specified MPLS next hop. Statistics for a next hop must be enabled with the **mpls statistics next-hop** command before they can be baselined. By default, the next-hop counters are baselined at zero.
- Example

  host1#**baseline mpls next-hop**

- There is no **no** version.

baseline mpls tunnel

- Use to set a statistics baseline for the specified MPLS tunnel. By default, the tunnel counters are baselined at zero.
- Example

  host1#**baseline mpls tunnel tunnel5**

- There is no **no** version.

mpls statistics label

- Use to enable statistics collection for MPLS in labels.
- By default, statistics are enabled for incoming labels and RSVP-TE or LDP outgoing labels, but not for others such as BGP outgoing labels. Statistics are not stored in NVS.
- Example

  host1#**mpls statistics label**

- Use the **no** version to disable statistics collection.

mpls statistics next-hop

- Use to enable statistics collection for MPLS next hops.
- By default, statistics are enabled for next hops depending on the protocol that created the MPLS next hop. Statistics are not stored in NVS.
- Example

  host1#**mpls statistics next-hop 1046**

- Use the **no** version to disable statistics collection.

mpls statistics policy

- Use to enable statistics collection for policies attached to an MPLS tunnel.
- Statistics are not stored in NVS.
- Example

  host1#**mpls statistics policy tunnel5**

- Use the **no** version to disable statistics collection.

show atm vc

- Use to display a summary of all configured ATM VCs and reserved VC ranges.
- If you are using the interface label space for labels, the display includes ATM VCs used as MPLS LSPs and VPI/VCI ranges reserved for MPLS.
- You can specify one or more of the following keywords individually or in combination:
  - **vpi**—Displays VCs on a specific VPI
  - **category**—Displays VCs that have a specific service category
  - **status**—Displays VCs with a certain status
- You can also specify the **reserved** keyword with no other keywords to display only a summary of all reserved VC ranges on the router. This includes VPI/VCI ranges reserved for use by MPLS.
- Field descriptions
  - Interface—Interface type and number
  - VPI—Virtual path identifier
  - VCI—Virtual channel identifier
  - VCD—Virtual circuit descriptor
  - Type—Type of circuit: PVC
  - Encap—Encapsulation method: AUTO, AAL5, MUX, SNAP, ILMI, F4-OAM
  - Category—Service type configured on the VC: UBR, UBR-PCR, NRT-VBR, RT-VBR, CBR
  - Rx/Tx Peak—Peak rate in Kbps
  - Rx/Tx Avg—Average rate in Kbps
  - Rx/Tx Burst—Maximum number of cells that can be burst at the peak cell rate
  - Status—State of the virtual circuit: Up, Down
  - Start VPI—Starting virtual path identifier (inclusive) of the reserved VC range
  - Start VCI—Starting virtual circuit identifier (inclusive) of the reserved VC range
  - End VPI—Ending virtual path identifier (inclusive) of the reserved VC range
  - End VCI—Ending virtual circuit identifier (inclusive) of the reserved VC range

- Example 1—Displays all VCs and reserved VC ranges on the router

```
host1#show atm vc
                                          Cate Rx/Tx Rx/Tx Rx/Tx Sta
Interface    VPI VCI  VCD  Type Encap gory Peak   Avg  Burst tus
------------ --- ---- ---- ---- ----- ---- ----- ----- ----- ---
ATM 3/0.2      0 101  4375 PVC  AUTO  CBR   1000     0     0 UP
ATM 3/0.3      0 102  4376 PVC  AUTO  CBR   1000     0     0 DOWN
...
ATM 3/0.8099   1 8099 8099 PVC  SNAP  UBR      0     0     0 UP
ATM 3/0.8100   1 8100 8100 PVC  SNAP  UBR      0     0     0 DOWN
8000 circuit(s) found


Reserved VCC ranges:

          Start Start End End
Interface  VPI   VCI  VPI VCI
---------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

- Example 2—Displays all reserved VC ranges on the router

```
host1#show atm vc reserved
Reserved VCC ranges:

          Start Start End End
Interface  VPI   VCI  VPI VCI
---------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

show cac

- Use to display global CAC (call admission control) configuration.
- Example

```
host1#show cac
resource info flood interval 180
```

show cac interface

- Use to display all interfaces on which traffic engineering bandwidth accounting is configured, or information only for the specified interface.
- Use the **brief** keyword to display summary information.
- Field descriptions
  - bandwidth—Maximum physical bandwidth in Kbps; line rate
  - IP/MPLS reserveable bw—Total bandwidth in Kbps that can be reserved for MPLS; includes bandwidth that is already reserved as well as bandwidth not yet reserved
  - current total available bw—Total bandwidth in Kbps that is available to be reserved

- MPLS TE flooding threshold up/down—Absolute percentages of total reservable bandwidth that trigger the flooding of the new bandwidth value throughout the network; flooding is triggered when bandwidth increases past any of the up threshold values and when bandwidth decreases past any of the down threshold values
- MPLS TE administrative weight—Weight assigned to the interface that supersedes a weight assigned by the IGP
- MPLS TE attribute flags—32-bit value that assigns the interface to a resource class and enables a tunnel to discriminate among interfaces by matching against tunnel affinity bits
- Available BW at 8 priority levels—Bandwidth in Kbps that is available at each priority level in the range 0–7

- Example

```
host1#show cac interface
atm2/0
  bandwidth 10 kbps
  IP/MPLS reserveable bw 10 kbps
  current total available bw 10 kbps
  MPLS TE flooding threshold:
    up   15 30 45 60 75 80 85 90 95 96 97 98 99 100
    down 100 99 98 97 96 95 90 85 80 75 60 45 30 15
  MPLS TE administrative weight 0
  MPLS TE attribute flags 0
  Available BW at 8 priority levels:
      0     10 kbps
      1     10 kbps
      2     10 kbps
      3     10 kbps
      4     10 kbps
      5     10 kbps
      6     10 kbps
      7     10 kbps
```

show configuration

- Use to display the configuration of all virtual routers or a specific virtual router.
- Example

```
host1#show configuration virtual-router euro7
```

show ip tunnel-route
show ipv6 tunnel-route

- Use to display the current state of the IPv4 or IPv6 tunnel routing table.
- You can display all routes, a specific route, best route to a resolved domain name, all routes beginning with a specified address, and routes for a particular protocol.
- Field descriptions
  - Prefix—IPv4 or IPv6 address prefix of network destination
  - Length—Network mask length for prefix
  - Type—Type of route; protocol

- Next Hop—IP address of the next hop to the route, whether it is a local interface or another router; not displayed for IPv6 tunnel routing table
- Dst or Distance—Administrative distance for the route
- Met or Metric—Number of hops; metric
- Interface—Interface type and interface specifier
- Tag—Numeric tag that identifies route
- Class—Attribute of a route applied only as a result of **set route-class** clause in a table map

- Example 1

```
host1:vr2#show ip tunnel-route all
Protocol/Route type codes:
  I1- ISIS level 1, I2- ISIS level2,
  I- route type intra, IA- route type inter, E- route type external,
  i- metric type internal, e- metric type external,
  O- OSPF, E1- external type 1, E2- external type2,
  N1- NSSA external type1, N2- NSSA external type2
  L- MPLS label, V- VRF, *- via indirect next-hop

  Prefix/Length      Type       Next Hop      Dst/Met                  Interface
------------------ --------- --------------- ---------- ----------------------------------------
200.200.200.1/32   Ldp       111.111.1.1[L18 110/2      ATM5/1.1
                             ]
                   Rsvp      200.200.200.1[L 110/2      ATM5/1.1
                             25]
```

- Example 2

```
host1:vr2#show ip tunnel-route 200.200.200.1/32 detail
Protocol/Route type codes:
  I1- ISIS level 1, I2- ISIS level2,
  I- route type intra, IA- route type inter, E- route type external,
  i- metric type internal, e- metric type external,
  O- OSPF, E1- external type 1, E2- external type2,
  N1- NSSA external type1, N2- NSSA external type2
  L- MPLS label, V- VRF, *- via indirect next-hop

200.200.200.1/32 Type: Ldp Distance: 110 Metric: 2 Tag: 0 Class: 0
  MPLS next-hop: 3, label 18 on ATM5/1.1 (ip19000003.mpls.ip), nbr 111.111.1.1
```

- Example 3

```
host1:pe1:pe11#show ipv6 tunnel-route all
Protocol/Route type codes:
  O- OSPF, E1- external type 1, E2- external type2,
  N1- SSA external type1, N2- NSSA external type2
  L- MPLS label, V- VRF, *- via indirect next-hop

         Prefix/Length             Type     Dst/Met                  Interface
------------------------------- --------- ---------- ----------------------------------------
::21.21.21.0/126                BgpTunnel 200/0      [L20,L26] ATM5/0.10
                                BgpTunnel 200/0      [L20,L34] ATM5/0.10
2::2/128                        BgpTunnel 200/0      [L20,L26] ATM5/0.10
                                BgpTunnel 200/0      [L20,L34] ATM5/0.10
```

- Example 4

```
host1:pe1:pe11#show ipv6 tunnel-route ::21.21.21.0/126 detail all
Protocol/Route type codes:
  O- OSPF, E1- external type 1, E2- external type2,
  N1- SSA external type1, N2- NSSA external type2
  L- MPLS label, V- VRF, *- via indirect next-hop

::21.21.21.0/126 Type: BgpTunnel Distance: 200 Metric: 0 Class: 0
  MPLS next-hop: 18, label 20, VPN traffic, resolved by MPLS next-hop 13
  MPLS next-hop: 13, resolved by MPLS next-hop 34, peer ::ffff:2.2.2.2
  MPLS next-hop: 34, ECMP next-hop, leg count 2
  MPLS next-hop: 17, resolved by MPLS next-hop 16, peer 2.2.2.2
  MPLS next-hop: 16, primary(in use): label 26 on ATM5/0.10, secondary: resolved by MPLS next-hop 0
```

show ldp

- Use to display information about LDP.
- The **mpls** keyword is optional and is provided for compatibility with non–E-series implementations.
- Field descriptions
  - LSR ID—IP address of label-switched router
  - FEC Deaggregation—State of FEC deaggregation, on or off
  - Egress label—Type of label advertised for the LSR that is the egress router for the prefix, implicit null, explicit null, or a non-null label
  - Label distribution control mode—Label distribution control mode used by LDP for label distribution, independent control, or ordered control
  - LDP session retry—Configured values for the number of LDP session retry attempts and the retry interval
  - LDP session hold time—Configured value for the LDP session hold time
  - LDP session keepalive interval—Interval at which LDP sends session keepalive messages, in seconds
  - LDP targeted-hello hold time—LDP targeted-hello hold time, in seconds
  - LDP targeted-hello interval—LDP targeted-hello interval, in seconds
  - Topology Driven LSP—Status of topology-driven LSP, enabled or disabled
  - LSPs used for IP forwarding—LSPs are placed in the IP routing table for forwarding plain IP traffic; displayed only when the **mpls ldp ip-forwarding** command has been configured. Indicates whether the LSPs that are used for IP forwarding are host only, subject to a specified access list, or subject to a specified prefix list.
  - LDP proto stats—LDP protocol statistics
    - totalPeersDiscovered—Number of LDP peers discovered
    - totalAdjacenciesEstablished—Number of LDP adjacencies established
    - totalSessionsEstablished—Number of LDP sessions established
    - totalFECElements—Number of FEC elements
    - totalFECs—Number of FECs

- totalInLabels—Number of in labels (sent to upstream neighbor)
- totalOutLabels—Number of out labels (received from downstream neighbor)
- totalCrLSPSetup—Number of constraint-based routed LSPs set up
- totalCrLSPDeleted—Number of constraint-based routed LSPs deleted

- Example

```
host1#show ldp
LDP
  LSR ID is 80.0.0.2
  FEC Deaggregation is off
  Egress label: implicit-null
  Label distribution control mode: ordered control
  LDP session retry  0 times at interval 10
  LDP session hold time: 180
  LDP session keepalive interval: 20
  LDP targeted-hello hold time: 45
  LDP targeted-hello interval: 15
  Topology Driven LSP enabled
  LSPs used for IP forwarding
      for host addresses only
```

show ldp binding
show mpls binding

- Use to display label bindings from the MPLS label information base. You can use either the **ldp** keyword or the **mpls** keyword to display the same information with this command.
- Field descriptions
  - In—Label sent to upstream neighbor for displayed route
  - Out—Label received from downstream neighbor for displayed route
  - neighbor—IP address of neighbor to which the label is sent or received
  - stale—Label that indicates neighbor has restarted
- Example

```
host1#show mpls binding

   Frame Relay over MPLS  vc-id 50001 group-id 2
    In    26 neighbor 10.9.1.3
    Out   27 neighbor 10.9.1.3
   VLAN over MPLS  vc-id 240001 group-id 2
    In    22 neighbor 10.9.1.3
    Out   25 neighbor 10.9.1.3

  10.1.1.1/32
    In    10001 neighbor 10.3.11.2
    Out   20001 neighbor 10.3.11.2

  10.2.2.2/32
    In    10002 neighbor 10.4.12.2    stale
    Out   20002 neighbor 10.4.12.2    stale
```

```
10.3.3.3/32
  In    10005 neighbor 10.4.12.2    stale
  Out   20003 neighbor 10.4.12.2    stale

10.4.12.0/30
  In    10003 neighbor 10.5.5.2
  Out   20004 neighbor 10.5.5.2

10.4.23.0/30
  In    10004 neighbor 10.5.5.2
  Out   20005 neighbor 10.5.5.2
```

show ldp graceful-restart

- Use to display information about LDP graceful restart.
- The **mpls** keyword is optional and is provided for compatibility with non–E-series implementations.
- Field descriptions
  - LDP Graceful Restart—State of graceful restart, enabled or disabled
  - Helper Mode—State of graceful restart helper mode, enabled or disabled
  - Reconnect Time—Locally configured value for reconnect time, in seconds
  - Recovery Time—Locally configured value for recovery time, in seconds
  - Max Recovery Time—Locally configured value for max-recovery timer, in seconds
  - Neighbor Liveness Timer—Locally configured value for neighbor-liveness timer, in seconds
  - Peer—Address, state, and LDP graceful restart state for neighbor
- Example

```
host1#show ldp graceful-restart
LDP Graceful Restart is enabled
Helper Mode is enabled
Reconnect Time: 220 sec
Recovery Time: 240 sec
Max Recovery Time: 260 sec
Neighbor Liveness Timer: 280 sec
  Peer 80.0.1.1:0, State: operational, Restarter Mode: disabled, Helper
Mode: enabled
  Peer 80.0.3.3:0, State: operational, Restarter Mode: disabled, Helper
Mode: enabled
```

show ldp igp-sync

- Use to display information about interfaces that are synchronizing with LDP or the specified interface that is synchronizing with LDP.
- Field descriptions
  - LDP—State of LDP, configured, auto-configured, or not configured
  - SYNC status—State of synchronization, enabled or disabled
  - IGP holddown time—Value of IGP hold down time, infinite or number of milliseconds

- Peer LDP Ident—IP address of LDP peer
- IGP enabled—IGP protocol

- Example

```
host1#show ldp igp-sync
Atm 0/0:
       LDP configured;  SYNC enabled.
       SYNC status: sync achieved; peer reachable.
       IGP holddown time: infinite.
       Peer LDP Ident: 10.130.0.1:0
       IGP enabled: OSPF 1
```

show ldp interface

- Use to display information about all LDP interfaces or the specified LDP interface.
- The **mpls** keyword is optional and is provided for compatibility with non–E-series implementations.
- Use the **brief** keyword to display a brief summary of interface information.
- Field descriptions
  - Interface—Identifier of the interface
  - autoconfigured—LDP has been autoconfigured on the interface
  - Interface address—IP address of the interface, with address mask
  - Enabled with profile—Name of profile with which interface was enabled
  - Configured hold time—Configured period for which a sending LSR maintains a record of link hello messages from the receiving LSR without receipt of another link hello message from that LSR, in seconds
  - Hello interval—Negotiated interval between link-hello packets, in seconds
  - Hold time—Lowest configured hold time among all neighbors on the same subnet, used as the effective hold time, in seconds
  - Number of adjacencies—Number of LDP adjacencies for the interface
  - Link hello adjacency—Address and transport address of the link hello adjacency; time adjacency has been up in *hh*:*mm*:*ss*; remaining hold time for the adjacency in seconds
  - Session statistics
    - label alloc—Number of labels allocated and advertised to this peer
    - label learned—Number of labels received from this peer
    - accum label alloc—Cumulative total number of labels allocated and advertised to this peer
    - accum label learned—Cumulative total number of labels received from this peer
    - last restart time—Time in *hh*:*mm*:*ss* since session last restarted
    - notf—Number of notification messages received or received bad or sent
    - msg—Number of messages received or received bad or sent

- mapping—Number of label mapping messages received or received bad or sent
- request—Number of label request messages received or received bad or sent
- abort—Number of label abort messages received or received bad or sent
- release—Number of label release messages received or received bad or sent
- withdraw—Number of label withdraw messages received or received bad or sent
- addr—Number of address messages received or received bad or sent
- addr withdraw—Number of address withdraw messages received or received bad or sent
- msgId—Number of message ID messages received or sent
- unknown msg type err—Number of unknown message type errors received

- Adjacency statistics
  - hello recv—Number of hello messages received
  - hello sent—Number of hello messages sent
  - bad hello recv—Number of hello messages received bad
  - adj setup time—Time in *hh*:*mm*:*ss* since adjacency set up
  - last hello recv time—Time in *hh*:*mm*:*ss* since last hello message received
  - last hello sent time—Time in *hh*:*mm*:*ss* since last hello message sent
  - remaining hold time—Time in *hh*:*mm*:*ss* remaining of the hold time
- IP-Address—IP address of the interface
- Protocol—Administrative state of LDP, enabled or disabled

- Example 1

```
host1#show ldp interface
Interface ATM6/0.120
    Interface address: 192.168.12.1/28
    Enabled with profile 'default'
    Configured hold time: 15
    Hello interval: 5
    Hold Time: 1
    242 hello received, 242 hello sent, 0 hello rejected
    1 adjacency created, 0 adjacency deleted,
  Number of adjacencies = 1
    Link hello adjacency: Address: 10.10.12.2, Transport address: 80.0.2.2,
      Up for 00:20:09, Remaining hold time: 11 sec
```

- Example 2—when LDP is autoconfigured

```
host1#show ldp interface
Interface FastEthernet0/0 (auto-configured)
    Interface address: 10.60.1.1/24
    Enabled with profile 'default'
    Configured hold time: 15
    Hello interval: 5
    Hold Time: 5
    60 hello received, 60 hello sent, 0 hello rejected
    1 adjacency created, 0 adjacency deleted,
  Number of adjacencies = 1
    Link hello adjacency: Address: 12.60.1.2, Transport address: 12.60.1.2,
      Up for 00:04:56, Remaining hold time: 14 sec
```

- Example 3

```
host1#show ldp interface brief
Interface              IP-Address          Protocol
ATM6/1.1               192.168.100.21/30   enabled
ATM6/1.3               192.168.100.17/30   enabled
ATM6/1.5               192.168.100.13/30   enabled
ATM6/0.7               172.16.100.1/30     enabled
ATM6/0.8               172.16.100.22/30    enabled
ATM6/0.9               172.16.100.14/30    enabled
```

show ldp neighbor

- Use to display LDP neighbor information.
- The **mpls** keyword is optional and is provided for compatibility with non–E-series implementations.
- If a password is configured for a peer, you can view the password with the **show configuration** command. This command displays the passwords in cleartext unless the **service password-encryption** command has been issued, in which case the passwords are displayed in encrypted format.
- You can issue the **brief** keyword to display only brief information about the LDP neighbors.
- You can issue the **graceful-restart** keyword to display information about graceful restart for neighbors; other detailed information about the neighbors is omitted.
- You can issue the **statistics** keyword to display information about LDP statistics for the session with each LDP neighbor
- Field descriptions
  - LDP neighbor—IP address of LDP peer
  - LSR—IP address of remote and local peers; the number following the colon is the platform label space ID, and is always 0
  - Transport address—Transport remote and local address for the TCP session
  - State—State of the session, nonexistent (session connection not established) or operational (has received keepalive message)
  - LDP advertisement—Mode of label distribution, downstream-unsolicited or downstream-on-demand
  - Up—Time that the adjacency has been up, in *hh*:*mm*:*ss* format

- Graceful Restart—State of graceful restart, enabled or disabled
- Helper Mode—State of graceful restart helper mode, enabled or disabled
- Reconnect Time—Value for reconnect time received from peer in FT TLV, in milliseconds
- Recovery Time—Value for recovery time received from peer in FT TLV, in milliseconds
- State—Status of the neighbor's graceful restart, one of the following:
  - nonexistent—LDP session is not established
  - restarting—Neighbor LSR is restarting
  - recovering—Session with neighbor LSR has reestablished after the neighbor restarted; label bindings are being exchanged
  - operational—LDP session is up
- Neighbor—IP address of LDP peer
- Initialization—Number of initialization messages received and sent
- Keepalive—Number of keepalive messages received and sent
- Notification—Number of notification messages received and sent
- Address—Number of address messages received and sent
- Address withdraw—Number of address withdraw messages received and sent
- Label mapping—Number of label mapping messages received and sent
- Label request—Number of label request messages received and sent
- Label withdraw—Number of label withdraw messages received and sent
- Label release—Number of label release messages received and sent

- Example 1

```
host1#show ldp neighbor 10.3.5.1
LDP Neighbor: 10.0.2.2
  LSR: Remote 10.0.2.2:0, local 10.0.1.1:0
  Transport address: remote 10.0.2.2, local 10.0.1.1
  State: Operational
  LDP advertisement: Unsolicited
  Up for 00:20:03

  Number of next-hop addresses received = 3
    10.0.2.2  100.6.12.2  100.6.23.2
  Number of adjacencies = 1
    Link Hello adjacency: address 10.6.12.2, transport 10.0.2.2,
      Up for 00:20:09, remaining hold time: 11 sec
```

- Example 2

```
host1#show ldp neighbor brief

Neighbor                 Transport Address   State
10.0.2.2                10.0.1.1->80.0.2.2     Operational
```

- Example 3

```
host1#show ldp neighbor graceful-restart
LDP Neighbor: 10.0.1.1
  Graceful Restart is disabled
  Helper Mode is enabled
  Reconnect Time: 0 msec
  Recovery Time: 0 msec
  State: operational

LDP neighbor 10.0.2.2
  Graceful Restart is enabled
  Helper Mode is enabled
  Reconnect Time: 220000 msec
  Recovery Time: 0 msec
  State: operational
```

- Example 4

```
host1#show ldp neighbor statistics

LDP Neighbor: 10.0.2.2
  Message type     Received   Sent
----------------   --------   ----
Initialization            1      1
Keepalive                85     85
Notification              0      0
Address                   1      1
Address withdraw          0      0
Label mapping             5      5
Label request             0      0
Label withdraw            2      2
Label release             2      2
```

show ldp profile

- Use to display a specific LDP profile, or all LDP profiles.
- The **mpls** keyword is optional and is provided for compatibility with non–E-series implementations.
- Field descriptions
  - profile—Number of interfaces that use the profile
  - session retry—Number of attempts that will be made to set up an MPLS LDP session
- Example

```
host1:pe2#show ldp profile default
ldp profile default: used by 2 interfaces
   session retry:  10 times at interval 10
```

show ldp statistics

- Use to display statistics for LDP on the current virtual router
- The **mpls** keyword is optional and is provided for compatibility with non–E-series implementations.

- Field descriptions
  - Hello—Number of hello messages received and sent
  - Initialization—Number of initialization messages received and sent
  - Keepalive—Number of keepalive messages received and sent
  - Notification—Number of notification messages received and sent
  - Address—Number of address messages received and sent
  - Address withdraw—Number of address withdraw messages received and sent
  - Label mapping—Number of label mapping messages received and sent
  - Label request—Number of label request messages received and sent
  - Label withdraw—Number of label withdraw messages received and sent
  - Label release—Number of label release messages received and sent
  - Label abort—Number of label abort messages received and sent
  - All UDP—Number of UDP messages received and sent
  - All TCP—Number of TCP messages received and sent
  - Sessions opened—Number of session opened events
  - Sessions closed—Number of session closed events
  - Topology changes—Number of topology change events
  - No router id—Number of no router ID events
  - No address—Number of no address events
  - No interface—Number of no interface events
  - No session—Number of no session events
  - No adjacency—Number of no adjacency events
  - Unknown version—Number of unknown version events
  - Malformed PDU—Number of malformed PDU events
  - Malformed message—Number of malformed message events
  - Unknown message type—Number of unknown message type events
  - Inappropriate message—Number of inappropriate message events
  - Malformed tlv—Number of inappropriate message events
  - Bad TLV value—Number of bad TLV value events
  - Missing TLV—Number of missing TLV events
  - PDU too large—Number of PDU too large events
  - PDU too small—Number of PDU too small events
  - No Memory—Number of no memory events

- Example

```
host1#show ldp statistics

  Message type     Received   Sent
-----------------  --------   -----
Hello                25733   25735
Initialization           2       2
Keepalive             9646    9646
Notification             0       0
Address                  2       2
Address withdraw         0       0
Label mapping            8       8
Label request            0       0
Label withdraw           0       0
Label release            0       0
Label abort              0       0
All UDP              25733   25735
All TCP               9654    9654

     Event type          Total
---------------------   -----
Sessions opened             2
Sessions closed             0
Topology changes            5
No router id                0
No address                  0
No interface                0
No session                  0
No adjacency                0
Unknown version             0
Malformed PDU               0
Malformed message           0
Unknown message type        0
Inappropriate message       0
Malformed tlv               0
Bad TLV value               0
Missing TLV                 0
PDU too large               0
PDU too small               0
No Memory                   0
```

show ldp targeted session

- Use to display LDP targeted hello receive or send list, or both.
- Field descriptions
  - D—Targeted session created by layer 2 over MPLS connection; see *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy*, for more information about layer 2 over MPLS
  - S—Targeted session statically created by user
  - A—Targeted session created by access list
  - Used By—Letter representing source of targeted session

- Example

```
host1#show ldp targeted session
Mpls Target Session Status:

 D = Dynamically, S = Statically, A = Access List Configured

 D = Dynamically, S = Statically, A = Access List Configured

    Targeted session sent to 10.9.1.3 is up  Used By: D
      indirect nexthop index 3, resolved
    Targeted session sent to 10.9.1.6 is up  Used By: S
      indirect nexthop index 206, resolved
```

show mpls

- Use to display status and configuration information about MPLS.
- Field descriptions
  - MPLS—Status of MPLS, administratively enabled or disabled, and configuration status
  - LSR ID—IP address of label-switched router
  - Re-optimization timer—Frequency at which LSPs are checked for better paths
  - Label range—Range of platform label space
  - retry—Retry behavior to be performed during LSP setup
  - Loop Detect—Status of loop detection, enabled or disabled
  - LDP—This field and the following fields are displayed only when LDP is enabled.
    - LSR ID—IP address of label-switched router
    - FEC Deaggregation—State of FEC deaggregation, on or off
    - Egress label—Type of label advertised for the LSR that is the egress router for the prefix, explicit-null or a non-null label
    - Label distribution control mode—Label distribution control mode used by LDP for label distribution, independent control or ordered control
    - LDP session retry—Interval in seconds between attempts to set up an MPLS LDP session
    - LDP session hold time—Period in seconds for which an LSR maintains the session with its LDP peer without receipt of any LDP message from that peer
    - LDP session keepalive interval—Interval at which LDP sends session keepalive messages, in seconds
    - LDP targeted hello hold time—LDP targeted-hello hold time, in seconds
    - LDP targeted-hello interval—LDP targeted-hello interval, in seconds

- Topology Driven LSP—Status of topology-driven LSP, enabled or disabled
- LSPs used for IP forwarding—LSPs are placed in the IP routing table for forwarding plain IP traffic; displayed only when the **mpls ldp ip-forwarding** command has been configured. Indicates whether the LSPs that are used for IP forwarding are host only, subject to a specified access list, or subject to a specified prefix list.

- RSVP is enabled—This field and the following fields are displayed only when RSVP-TE is enabled.
  - LSRID—IP address of label-switched router
  - Re-optimization timer—Frequency at which LSPs are checked for better paths
  - Tunnel retry—Retry behavior to be performed during LSP setup
  - Refresh reduction—State of RSVP-TE summary refresh reduction, OFF or ON
  - Message bundling—State of RSVP-TE summary refresh message bundling, OFF or ON
  - Egress label—Type of label advertised for the LSR that is the egress router for the prefix, explicit-null or a non-null label
  - Hellos—State of RSVP-TE hello feature, including the hello refresh interval and hello miss limit
  - Graceful restart—State of RSVP-TE graceful restart, OFF or ON
  - helper mode—Graceful restart helper mode is enabled when this field is displayed
  - Restart time—Graceful restart time, in milliseconds
  - Recovery time—Graceful restart recovery time, in milliseconds

- Example 1

```
host1#show mpls
MPLS administratively enabled
  Current state is Config incomplete
  LSR ID is 10.2.2.2
  Re-optimization timer is 3600
  Label range 3000 ~ 4000
  retry forever at interval 30 during LSP setup if there is route
  retry forever at interval 30 during LSP setup if there is no route
Loop Detect enabled
```

- Example 2—Additional detail is shown when LDP is enabled.

```
LDP
  LSR ID is 80.0.0.2
  FEC Deaggregation is off
  Egress label: implicit-null
  Label distribution control mode: ordered control
  LDP session retry  0 times at interval 10
  LDP session hold time: 180
  LDP session keepalive interval: 20
  LDP targeted-hello hold time: 45
  LDP targeted-hello interval: 15
```

```
  Topology Driven LSP enabled
  LSPs used for IP forwarding
      for host addresses only
```

- Example 3—Additional detail is shown when RSVP-TE is enabled.

```
RSVP is enabled
  LSRID 10.1.1.1
  Re-optimization timer is 3600
  Tunnel retry forever at interval 5 if route is available
  Tunnel retry forever at interval 5 if no route is available
  Refresh reduction is OFF
  Message bundling is OFF
  Egress label is non-null
  Hellos are on with an interval of 10000 and miss limit of 4
  Graceful restart is ON
    Restart time 60000 milliseconds
    Recovery time 120000 milliseconds
```

- Example 4—Shows RSVP-TE graceful restart helper mode.

```
RSVP is enabled
...
  Graceful restart is ON (helper mode)
```

show mpls explicit-paths

- Use to display all explicit paths or a particular explicit path.
- Field descriptions
  - path name/identifier—Name or identifier of explicit path and status, enabled or disabled, followed by list of path links and the IP address for each link's next address
- Examples

```
host1:pe2#show mpls explicit-paths
  path name/identifier rx1-path enabled
    1: next-address 70.70.70.2
    2: next-address 30.30.30.1
  not referenced by any options
  path name/identifier rx1-path2 enabled
    1: next-address 60.60.60.2
    2: next-address 40.40.40.1
  not referenced by any options

host1:pe2#show mpls explicit-paths name rx1-path2
  path name/identifier rx1-path2 enabled
    1: next-address 60.60.60.2
    2: next-address 40.40.40.1
  not referenced by any options
```

show mpls fast-reroute database

- Use to display information about the backup status of protected primary LSPs.
- Field descriptions
  - Role—Role of the router in the LSP: core, head, or tail
  - Name—Name of the primary LSP
  - OutIntf / Label—Interface type and specifier of the outgoing interface, and the label associated with that interface

  - BackupIntf / Label—Interface type and specifier of the backup interface, and the label associated with that interface
  - Backup Status—Status of backup protection (bypass) for the LSP
- Example on a core router

```
host1(config-if)#show mpls fast-reroute database
                          OutIntf                                  Backup
Role        Name          / Label       BackupIntf / Label         Status
----   ---------------   -------------   ----------------------   -----------
Core   LSP 10.1.1.1:6    ATM4/0.2 / 21   tun mpls:bypass23 / 21   Established
Core   LSP 10.1.1.1:7    ATM4/0.2 / 26   tun mpls:bypass23 / 26   Established

Example on a tunnel ingress router
                OutIntf                                  Backup
Role   Name     / Label       BackupIntf / Label        Status
----   ----   -------------   ----------------------   -----------
Head   1      ATM4/0.1 / 27   tun mpls:bypass12 / 27   Established
Head   p2     ATM4/0.1 / 21   tun mpls:bypass12 / 21   Established
```

show mpls forwarding

- Use to display information for labels being used for forwarding.
- Field descriptions
  - In label—Label sent to upstream neighbor for route
  - Out label—Label received from downstream neighbor for route
  - Label space—Label space in which the label is assigned
  - Owner—Signaling protocol that placed the label in the forwarding table: BGP, LDP, or RSVP-TE
  - Spoof check—Type and location of spoof checking performed on the MPLS packet, router or interface
  - Action—Action taken for MPLS packets arriving with that label
  - in pkts—Number of packets sent with the label
  - in Octets—Number of octets sent with the label
  - in errors—Number of packets that are dropped for some reason before being sent
  - in discardPkts—Number of packets that are discarded due to lack of buffer space before being sent
- Example 1

```
host1:vr2#show mpls forwarding
In label: 28
  Label space: platform label space
  Owner: ldp
  Spoof check: router pe1
  Action:
    MPLS next-hop: 1, lookup on inner header/label
  Statistics:
    0 in pkts
    0 in Octets
    0 in errors
    0 in discard pkts
```

- Example 2

```
host1:vr2#show mpls forwarding brief
Platform label space

In Label  Owner                              Action
--------- -------- ---------------------------------------
28        ldp      lookup on inner header/label
29        ldp      swap to 20 on ATM2/0.10, nbr 10.10.10.2
30        ldp      lookup on inner header/label
31        ldp      swap to 22 on ATM2/0.10, nbr 10.10.10.2
32        ldp       swap to 23 on ATM2/0.10, nbr 10.10.10.2
```

show mpls interface

- Use to display status and configuration information about MPLS interfaces.
- Use the **shim** keyword to display information about shim interfaces (used for layer 2 over MPLS). Use the **minor** keyword to display information about minor interfaces. Use the **state not-up** keyword to display information about interfaces that are not in the up state.
- Field descriptions
  - Interface—Specifier and status of each interface
  - RSVP—Status of RSVP, configured or not, and profile used
  - LDP—Status of LDP, configured or not configured, and profile used
  - IP interfaces on this MPLS interface—IP address of IP interfaces and session status
  - Condensed location—Internal, platform-dependent, 32-bit representation of the interface location, used by Juniper Networks Customer support for troubleshooting.
  - Session statistics
    - label alloc—Number of labels allocated and advertised to this peer
    - label learned—Number of labels received from this peer
    - accum label alloc—Cumulative total number of labels allocated and advertised to this peer
    - accum label learned—Cumulative total number of labels received from this peer
    - notf—Number of notification messages received or received bad or sent
    - mapping—Number of label mapping messages received or received bad or sent
    - msg—Number of messages sent or received
    - request—Number of label request messages received or received bad or sent
    - abort—Number of label request abort messages for downstream on demand received or received bad or sent
    - release—Number of label release messages received or received bad or sent

- withdraw—Number of label withdraw messages received or received bad or sent
- addr—Number of address messages received or received bad or sent
- addr withdraw—Number of address withdraw messages received or received bad or sent
- msgId—Number of message IDs received or sent
- unknown message type err—Number or unknown message type errors received
- last info error code—Last received notification code
- loop detected—Loop detected; for downstream on demand

- Adjacency statistics
  - hello recv—Number of hello messages received
  - hello sent—Number of hello messages sent
  - bad hello recv—Number of hello messages received bad
  - adj setup time—Time in *hh*:*mm*:*ss* since adjacency set up
  - last hello recv time—Time in *hh*:*mm*:*ss* since last hello message received
  - last hello sent time—Time in *hh*:*mm*:*ss* since last hello message sent
- MPLS Statistics
  - failed lbl lookup—Number of packets received whose labels are not recognized
  - octets—Number of octets received or sent
  - hcoctets—Number of high-capacity (64-bit) octets received or sent
  - pkts—Number of packets received or sent
  - hcpkts—Number of high-capacity (64-bit) packets received or sent
  - errors—Number of packets that are dropped for some reason at receipt or before being sent
  - discards—Number of packets that are discarded due to lack of buffer space at receipt or before being sent
  - adjacency—Number of adjacencies currently established
  - session—Number of sessions currently established
  - accum adjacency—Cumulative total number of adjacencies established since interface is up
  - accum session—Cumulative total number of sessions established since interface is up
  - hello recv—Number of hello messages received
  - hello sent—Number of hello messages sent
  - hello rej—Number of hello messages rejected
  - adj setup—Number of adjacencies set up
  - adj deleted—Number of adjacencies deleted

- Example 1—For all interfaces

```
host1:pe1#show mpls interface
MPLS major interface ATM2/0.10
  ATM circuit type is 1483 LLC encapsulation
  Administrative state is enabled
  Operational state is up
  Operational MTU is 9180
  Received:
    0 packets
    0 bytes
    0 errors
    0 discards
    0 failed label lookups
  Sent:
    0 packets
    0 bytes
    0 errors
    0 discards

LDP information:
    10.1.1.2/24
    enabled with profile 'default'
     0 hello recv, 1 hello sent, 0 hello rej
     0 adj setup, 0 adj deleted,

  RSVP
    Enabled with profile default
    Authentication is disabled
    Authentication key: <none>
    Hellos are on with an interval of 10000 and miss limit of 4
    Hello settings are not inherited

MPLS minor interface pe1-to-pe2 (transmit)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  Sent:
    0 packets
    0 bytes

  queue 0: traffic class best-effort, bound to atm-vc ATM2/0.10
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0


MPLS minor interface lsp-02020202-1-4 (receive)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  Statistics not enabled for this interface
```

- Example 2—RSVP configured and RSVP authentication enabled

```
host1:pe2#show mpls interface atm 5/1.1
Interface ATM5/1.1 Up
 RSVP enabled with profile default
   Authentication: enabled
   Authentication Key: <a password has been configured>
 LDP not configured
 IP interfaces on this MPLS interface:
    192.168.100.21/30
```

```
MPLS Statistics:
  Rcvd: 0 failed lbl lookup, 0 octets, 0 hcOctets
        0 pkts, 0 hcPkts, 0 errors, 0 discards
  Sent: 0 octets, 0 hcOctets, 0 pkts
        0 hcPkts, 0 errors, 0 discards
...
```

- Example 3—For a specific interface

```
host1:pe2#show mpls interface atm 2/0.60
Interface atm2/0.60 Up
 RSVP not configured
 LDP enabled with profile default
 IP interfaces on this MPLS interface:
    60.60.60.1/16  Session to 4.4.4.4 is operational (active)
Session statistics:
          12 label alloc, 12 label learned,
          12 accum label alloc, 12 accum label learned,
          last restart time = 00:04:44
          Rcvd: 0 notf, 29 msg, 12 mapping, 0 request
                0 abort, 0 release, 0 withdraw, 1 addr
                0 addr withdraw, 29 msgId
                0 bad mapping, 0 bad request, 0 bad abort, 0 bad release
                0 bad withdraw, 0 bad addr, 0 bad addr withdraw
                0 unknown msg type err
                last info err code = 0x00000000, 0 loop detected
          Sent: 0 notf, 29 msg, 12 mapping, 0 request
                0 abort, 0 release, 0 withdraw, 1 addr
                0 addr withdraw, 29 msgId
        Adjacency statistics:
          58 hello recv, 57 hello sent, 0 bad hello recv
          adj setup time = 00:04:44
          last hello recv time = 00:00:05, last hello sent time = 00:00:05
MPLS Statistics:
   Rcvd: 0 failed lbl lookup, 0 octets, 0 hcOctets
         0 pkts, 0 hcPkts, 0 errors, 0 discards
   Sent: 0 octets, 0 hcOctets, 0 pkts
         0 hcPkts, 0 errors, 0 discards
   1 adjacency, 1 session, 3 accum adjacency, 3 accum session
   14058 hello recv, 14063 hello sent, 0 hello rej
   3 adj setup, 2 adj deleted
```

- Example 4—Excerpt of output showing LDP and RSVP information displayed for an interface

```
host1:vr2#show mpls interface atm 6/1.1
...
MPLS major interface ATM6/1.1
  ATM circuit type is 1483 LLC encapsulation
  Administrative state is enabled
  Operational state is up
  Operational MTU is 9180
  Received:
    0 packets
    0 bytes
    0 errors
    0 discards
    0 failed label lookups
  Sent:
    0 packets
    0 bytes
    0 errors
    0 discards
```

```
LDP information:
    10.1.1.1/24
    enabled with profile 'default'
     0 hello recv, 2 hello sent, 0 hello rej
     0 adj setup, 0 adj deleted,

  RSVP
    Enabled with profile default
    Authentication is disabled
    Authentication key: <none>
    Hellos are on with an interval of 10000 and miss limit of 4
    Hello settings are not inherited
```

- Example 5—Detailed display of information

```
host1:pe1#show mpls interface detail
MPLS major interface ATM2/0.10
  ATM circuit type is 1483 LLC encapsulation
  Administrative state is enabled
  Operational state is up
  Operational MTU is 9180
  MPLS major interface UID is 0x19000001
  Lower interface UID is 0x0b000031
  Uses platform label space
  Peer IPv4 interface is ATM2/0.10 (UID 0x000000be)
  No peer IPv6 interface
  Upper IPv4 interface is ip19000001.mpls.ip (UID 0x000000bf, FEC index
0x0000003f)
  No upper IPv4 VPN interface
  No upper IPv6 interface
  No upper IPv6 VPN interface
  Condensed location is 0x00020000
  Received:
    0 packets
    0 bytes
    0 errors
    0 discards
    0 failed label lookups
  Sent:
    0 packets
    0 bytes
    0 errors
    0 discards
  RSVP
    Enabled with profile default
    Authentication is disabled
    Authentication key: <none>

MPLS minor interface pe1-to-pe2 (transmit)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  MPLS minor interface UID is 0x1a000001
  Lower MPLS major interface UID is 0x19000001
  Sent:
    0 packets
    0 bytes

  queue 0: traffic class best-effort, bound to atm-vc ATM2/0.10
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
```

```
MPLS minor interface lsp-02020202-1-4 (receive)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  MPLS minor interface UID is 0x1a000004
  Lower MPLS major interface UID is 0x19000001
  Statistics not enabled for this interface
```

- Example 6—Brief display of information

```
host1:pe1#show mpls interface brief

MPLS major interfaces

             Admin    Oper
Interface    state    state
---------   -------   -----
ATM2/0.10   enabled   up

MPLS shim interfaces

            Remote-PE   Virtual     Load
               or       Circuit   Balancing   Admin   Oper
Interface   LSP-name      ID        Group     state   state
---------   ---------   -------   ---------   -----   -----

MPLS minor interfaces

                     Lower     Oper
   Interface       MplsMajor   state   Direction
----------------   ---------   -----   ---------
pe1-to-pe2         ATM2/0.10   up      transmit
lsp-02020202-1-4   ATM2/0.10   up      receive
ERX-01-0c-d7:pe1#
```

- Example 7—Excerpt of detailed display showing RSVP-TE information

```
host1:vr2#show mpls interface detail
MPLS major interface fastEthernet6/0
RSVP
    Enabled with profile default
    Authentication is disabled
    Authentication key: <none>
    Hellos are enabled
    Hellos interval is 10000 milliseconds
    Hellos miss limit is 4
    Hello settings are not inherited
```

show mpls minor-interface

- Use to display status and configuration information about MPLS minor interfaces. You can display the same information with the **show mpls interface minor** command.
- Use the **state not-up** keyword to display information about interfaces that are not in the up state.
- Field descriptions
  - Interface—Specifier and status of each interface

- Example 1

```
host1:pe1#show mpls minor-interface
MPLS minor interface pe1-to-pe2 (transmit)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  Sent:
    0 packets
    0 bytes

  queue 0: traffic class best-effort, bound to atm-vc ATM2/0.10
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0


MPLS minor interface lsp-02020202-1-4 (receive)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  Statistics not enabled for this interface
```

- Example 2—Detailed display of minor interface information

```
host1:pe1#show mpls minor-interface detail
MPLS minor interface pe1-to-pe2 (transmit)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  MPLS minor interface UID is 0x1a000001
  Lower MPLS major interface UID is 0x19000001
  Sent:
    0 packets
    0 bytes

  queue 0: traffic class best-effort, bound to atm-vc ATM2/0.10
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0


MPLS minor interface lsp-02020202-1-4 (receive)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  MPLS minor interface UID is 0x1a000004
  Lower MPLS major interface UID is 0x19000001
  Statistics not enabled for this interface
```

- Example 3—Brief display of minor interface information

```
host1:pe1#show mpls minor-interface brief
                     Lower     Oper
   Interface       MplsMajor   state   Direction
----------------   ---------   -----   ---------
pe1-to-pe2         ATM2/0.10   up      transmit
lsp-02020202-1-4   ATM2/0.10   up      receive
ERX-01-0c-d7:pe1#
```

show mpls next-hop

- Use to display MPLS next hops and any available next-hop statistics.
- When one MPLS next-hop points to another MPLS next hop, the second next hop is displayed indented to the first one. This command also displays statistics for a next hop, if available.
- Next hops can be pointed to by MPLS forwarding entries on an LSR, IP or IPv6 routes on an LER, and VPLS bridge groups.
- Field descriptions
  - index—Next-hop index
  - label—Label for next hop
- Example

```
host1:vr2#show mpls next-hop
MPLS next-hop: index 1, lookup on inner header/label
   Statistics are not collected for MPLS switch-context next-hops

MPLS next-hop: index 2, lookup in router pe1
   Statistics are not collected for MPLS switch-context next-hops

MPLS next-hop: index 22, ECMP next-hop, leg count 2
   MPLS next-hop: index 20, label 36 on FastEthernet1/1.120, neighbor
10.120.120.1
   MPLS next-hop: index 21, label 36 on ATM2/1.20, neighbor 10.20.20.1
   Statistics are not collected for MPLS ECMP next-hops

MPLS next-hop: index 24, label 17, resolved by MPLS nextHop index 10
   MPLS next-hop: index 10, resolved by MPLS nextHop index 14, peer address
10.1.1.1
      MPLS next-hop: index 14, ECMP next-hop, leg count 2
         MPLS next-hop: index 12, label 32 on FastEthernet1/1.120, neighbor
10.120.120.1
         MPLS next-hop: index 13, label 32 on ATM2/1.20, neighbor 10.20.20.1
   Sent:
     0 packets
     0 bytes
     0 errors
     0 discards

MPLS next-hop: index 25, label 18, resolved by MPLS nextHop index 10
   MPLS next-hop: index 10, resolved by MPLS nextHop index 14, peer address
10.1.1.1
      MPLS next-hop: index 14, ECMP next-hop, leg count 2
         MPLS next-hop: index 12, label 32 on FastEthernet1/1.120, neighbor
10.120.120.1
         MPLS next-hop: index 13, label 32 on ATM2/1.20, neighbor 10.20.20.1
   Sent:
     0 packets
     0 bytes
     0 errors
     0 discards
```

show mpls phb-id

- Use to display the configured mapping between PHB IDs and traffic class/color combinations.
- PHB IDs used for L-LSPs do not have color.
- Field descriptions
  - phb-id—Per-hop behavior ID for which a traffic class/color combination is displayed
  - traffic-class—Traffic class associated with traffic
  - color—Color (drop precedence) associated with traffic, green, yellow, or red
- Example

```
host1#show mpls phb-id
Mpls PHB-ID traffic-class/color mappings:

--------   ------   --   -------------   -----------   -----   ------
standard   phb-id   0    traffic-class   best-effort   color   green
private    phb-id   0    traffic-class   best-effort   color   yellow
private    phb-id   1    traffic-class   1             color   red
private    phb-id   2    traffic-class   2             color   green
standard   phb-id   1    traffic-class   1             color   yellow
standard   phb-id   2    traffic-class   2             color   red
standard   phb-id   3    traffic-class   3             color   green
standard   phb-id   4    traffic-class   4             color   green
standard   phb-id   6    traffic-class   6             color   n/a
standard   phb-id   7    traffic-class   7             color   n/a
standard   phb-id   10   traffic-class   5             color   n/a
```

show mpls profile

- Use to display a specific RSVP-TE or tunnel profile, or all RSVP-TE or tunnel profiles.
- Field descriptions
  - profile—Number of interfaces that use the profile
  - refresh-period—Timeout period in seconds between generation of refresh messages
  - timeout factor—Number of refresh messages that can be lost before the session is ended
- Examples

```
host1:pe2#show mpls rsvp profile default
RSVP profile default: used by 0 interfaces
    refresh period: 30000 ms
    timeout factor: 3

host1#show mpls tunnels profile
MPLS Tunnel Profile tunnelProfile
  LSP setup using rsvp-te
  tunnel not announced to any IGP
  (Global) Retry forever
       at (Global) interval 5 during Lsp setup if there is route
  (Global) Retry forever
       at (Global) interval 5 during Lsp setup if there is no route
```

```
metric is relative 0
path option 2
  path to be dynamically calculated by isis
destinations include: 1.1.1.1 2.2.2.2 3.3.3.3
  ISIS Level 2 routers
  OSPF border routers
```

show mpls rsvp

- Use to display RSVP path state control blocks, reservation state control blocks, or session information about the virtual router to assist in debugging.
- Sessions can be any of the following:
  - ingress—Originating on the router
  - egress—Terminating on the router
  - transit—Travelling through the router
- Field descriptions
  - PSB—Path state control block
  - RSB—Reservation state control block
  - Sender—IP address of PSB or RSB sender
  - LSPId—ID of LSP
  - timeout—Period of time in milliseconds before PSB/RSB times out if no refresh arrives.
  - InLabel—Incoming label information
  - Associated tunnel—Tunnel identifier for minor interface for which the RSVP information is displayed
  - PHopIntf—Penultimate hop interface
  - IncomingIntf—Incoming interface
  - OutgoingIntf—Outgoing interface
  - PHopAddr—Penultimate hop address
  - m_ipNextHopAddr—Next hop address
  - NextHop—Type of next hop (loose, strict, session)
  - LabelRange—RSVP session label range
  - SenderTSpec—Traffic parameters for the sender
    - Token Bucket Rate—Sender's description of generated traffic, in kbps
    - Token Bucket Size—Sender's description of generated traffic, in kbps
    - Peak Data Rate—Lender's peak traffic generation rate
    - Min Policed Unit—Minimum packet size generated by sender
    - Max Packet Size—Maximum packet size generated by sender
  - RRO—Record route object
  - ADSPEC—Indicates presence of this QoS object
  - IN ERO—Incoming explicit route object
  - OUT ERO—Outgoing explicit route object

- SES ATTR—RSVP session attributes
  - Setup Pri—Setup priority of tunnel
  - Hold Pri—Hold priority of tunnel
  - name—Name of the tunnel
  - Flags—One or more of the IngressReRoute (the ingress router can reroute the LSP), Local Protection (routers can use local repair mechanism to fix the LSP; this fix might violate the explicit route object associated with the LSP), and MergingPermitted (LSPs can be merged) flags
- TTC—Indicates presence of the traffic trunk classifier object
- Policy Object—Indicates presence of the policy object
- Unknown Objects—Indicates presence objects not defined by the RSVP specification
- PSB Flags
  - InUse—PSB in use
  - Deleted—PSB deleted
  - NonRsvp—Non-RSVP hop present
  - RouteChangeNotify—Route change notification received
  - EroChanged—Explicit route object changed
  - NextHopChanged—Next hop has changed
  - RtNextHopChanged—Routing table next hop changed
  - EgressStatusChanged—PSB egress status has changed
  - QosChanged—QoS characteristics have changed
  - LabelChanged—Label has changed
  - ResvRefreshNeeded—Reservation refresh needed
  - PathRefreshNeeded—Path refresh needed
  - RroRequired—Record route object required
  - Egress—Session is egress
  - PathRefreshSent—Path refresh sent
  - EgressFilterFF—Egress filter reservation style Fixed Filter
  - QosCorrectionNeeded—QoS correction needed
  - IsPathTrigger—Has path refresh been triggered
- RSB Flags
  - InUse—RSB in use
  - Deleted—RSB deleted
  - RcvdAck—Acknowledgment received
  - StyleConverted— Reservation style converted to shared explicit
  - IsPathTrigger—Reservation refresh triggered
- Destination—RSVP session destination address

- TunnelId—Number representing the RSVP session tunnel ID
- Extended Tunnel Id—IP address representing the RSVP session extended tunnel ID

- Examples for an ingress session

```
host1#show mpls rsvp psb
  PSB: Sender 223.10.1.1 LSPId 1 timeout -- InLabel --
       PHopIntf
       IncomingIntf
       OutgoingIntf ATM2/0.1
       PHopAddr 0.0.0.0 m_ipNextHopAddr 221.1.1.1
       NextHop 221.1.1.1/255.255.255.255 (strict)
       LabelRange --
       SenderTSpec CType IntServ Controlled Load
                   Token Bucket Rate 0
                   Token Bucket Size 0
                   Peak Data Rate 0
                   Min Policed Unit 0
                   Max Packet Size 0
       Flags : InUse
               RroRequired
               PathRefreshSent

host1#show mpls rsvp rsb
  RSB: Timeout 157500 label 1/33
       Flags : InUse
               StyleConverted

host1:two#show mpls rsvp sessions
Destination 222.9.3.1 TunnelId 1 Extended Tunnel Id 223.10.1.1
  PSB: Sender 223.10.1.1 LSPId 1 timeout 157500 InLabel 17
       Associated Minor Interface: Tunnel 223.10.1.1:1
       PHopIntf ATM2/0.1
       IncomingIntf ATM2/1.1
       OutgoingIntf ATM2/0.3
       PHopAddr 221.1.1.2 m_ipNextHopAddr 122.1.1.1
       NextHop 122.1.1.1/255.255.255.255 (strict)
       LabelRange (generic) min 0 max 1048575
       SenderTSpec CType IntServ Controlled Load
                   Token Bucket Rate 0
                   Token Bucket Size 0
                   Peak Data Rate 0
                   Min Policed Unit 0
                   Max Packet Size 0
       RRO IPv4 hop 221.1.1.2 (strict)
       ADSPEC --
       IN ERO IPv4 hop 221.1.1.1 (strict)
              IPv4 hop 122.1.1.1 (strict)
       OUT ERO IPv4 hop 122.1.1.1 (strict)
       SES ATTR Setup Pri 4, Hold Pri 4, name --
                Flags : IngressReRoute
       TTC --
       Policy Object --
       Unknown Objects --
       Flags : InUse
               PathRefreshSent
```

```
RSB: Timeout 157500 label 16
     Associated Minor Interface: Tunnel 223.10.1.1:1
     FlowSpec CType IntServ Controlled Load
              Token Bucket Rate 0
              Token Bucket Size 0
              Peak Data Rate 0
              Min Policed Unit 0
              Max Packet Size 0
     RRO IPv4 hop 122.1.1.1 (strict)
     Policy Object --
     Unknown Objects --
     Flags : InUse
             StyleConverted
             ResvRefrSent
```

show mpls rsvp authentication

- Use to display information about RSVP MD5 authentication.
- For point-to-point interfaces, this command displays a single secure association with the single peer at the remote end. For multiaccess type interfaces (Ethernet), this command displays, if present, multiple secure associations from the various RSVP speakers.
- Field descriptions
  - RSVP Authentication Secure Association with peer—IP address of a peer with which the router has a security association
  - Receive Sequence Number—Sequence number of first authenticated packet from peer; subsequent packets from the peer must be greater than this base number
- Example

```
host1#show mpls rsvp authentication
Mpls interface FastEthernet2/4
  RSVP Authentication Secure Association with peer 10.2.2.2
    Receive Sequence Number 4592798942692985943
  RSVP Authentication Secure Association with peer 10.3.3.3
    Receive Sequence Number 4592798942692912623

Mpls interface ATM6/0.2
  RSVP Authentication Secure Association with peer 102.2.2.2
    Receive Sequence Number 4592798942692985934

Mpls interface ATM6/0.3
  RSVP Authentication Secure Association with peer 10.2.2.2
    Receive Sequence Number 4592798942692985956
```

show mpls rsvp bfd interfaces

- Use to display information about RSVP-TE major interfaces on which BFD is enabled.
- Field descriptions
  - Interface—RSVP-TE major interface on which BFD is enabled
  - Minimum Interval—Minimum interval in milliseconds, used when the minimum receive interval and minimum transmit intervals have the same value

- Minimum Rx-Interval—Minimum receive interval in milliseconds; minimum interval at which the local peer must receive BFD control packets from the remote peer
- Minimum Tx-Interval—Minimum transmit interval in milliseconds; interval at which the local peer proposes to transmit BFD control packets to the remote peer
- Multiplier—Detection multiplier value; roughly equivalent to the number of packets that can be missed before the BFD session is declared to be down

- Example

```
host1#show mpls rsvp bfd interfaces
                 Bfd Enabled RSVP interfaces
                 ---------------------------
             Minimum      Minimum       Minimum
Interface    Interval   Rx-Interval   Tx-Interval   Multiplier
---------    --------   -----------   -----------   ----------
ATM2/0.1     300        300           300           3
```

show mpls rsvp counters

- Use to display various counters for a particular RSVP interface or all RSVP interfaces.
- Field descriptions
  - Path Sent—Number of path messages sent on the interface
  - Path Rcvd—Number of path messages received on the interface
  - Path Error Sent—Number of patherror messages sent on the interface
  - Path Error Rcvd—Number of patherror messages received on the interface
  - Path Tear Sent—Number of pathtear messages sent on the interface
  - Path Tear Rcvd—Number of pathtear messages received on the interface
  - Resv Sent—Number of resv messages sent on the interface
  - Resv Rcvd—Number of resv messages received on the interface
  - Resv Error Sent—Number of resverr messages sent on the interface
  - Resv Error Rcvd—Number of resverr messages received on the interface
  - Resv Tear Sent—Number of resvtear messages sent on the interface
  - Resv Tear Rcvd—Number of resvtear messages received on the interface
  - Resv Conf Sent—Number of resvconf messages sent on the interface
  - Resv Conf Rcvd—Number of resvconf messages received on the interface
  - Srefresh Conf Sent—Number of srefresh messages sent on the interface
  - Srefresh Conf Rcvd—Number of srefresh messages received on the interface
  - Ack Conf Sent—Number of resvconf messages sent on the interface
  - Ack Conf Rcvd—Number of resvconf messages received on the interface
  - Nack Objects Sent—Number of nack objects sent on the interface
  - Nack Objects Rcvd—Number of nack objects received on the interface

- Msg Bundles Objects Sent—Number of message bundles sent on the interface
- Msg Bundles Rcvd—Number of message bundles received on the interface
- Error Msgs Rcvd—Number of error messages received on the interface
- Misordered Messages—Number of misordered messages received on the interface
- Send Failures—Number of failures to successfully send messages on the interface
- Path Triggers—Number of locally triggered path messages
- Resv Triggers—Number of locally triggered resv messages
- Forwarded Pkts—RSVP control packets that are forwarded through the router
- Hello Sent—Number of hello messages sent
- Hello Rcvd—Number of hello messages received
- Hello Ack Sent—Number of acknowledgments sent in response to hello requests received
- Hello Ack Rcvd—Number of acknowledgments received in response to hello requests sent
- Hello Discarded—Number of hello messages discarded
- Hello Ack Discarded—Number of hello ack messages discarded
- Hello Suppressed—Number of hello messages suppressed; message generation is suppressed when a hello request object is received from the destination node within the hello interval

- Example

```
host1#show mpls rsvp counters atm 6/0.1
Interface ATM6/0.1
  Path Sent           247           Path Rcvd           0
  Path Error Sent     0             Path Error Rcvd     0
  Path Tear Sent      0             Path Tear Rcvd      0
  Resv Sent           0             Resv Rcvd           245
  Resv Error Sent     0             Resv Error Rcvd     0
  Resv Tear Sent      0             Resv Tear Rcvd      0
  Resv Conf Sent      0             Resv Conf Rcvd      0
  SRefresh Sent       0             SRefresh Rcvd       0
  Ack Sent            0             Ack Rcvd            0
  Nack Objects Sent   0             Nack Objects Rcvd   0
  Msg Bundles Sent    0             Msg Bundles Rcvd    0
  Error Msgs Rcvd     0             Misordered Messages 0
  Send Failures       0             Msgs not acked      0
  Path Triggers       1             Resv Triggers       0
  Forwarded Pkts      0
  Hello Sent          7097          Hello Rcvd          7097
  Hello Ack Sent      0             Hello Ack Rcvd      7097
  Hello Discarded     0             Hello Ack Discarded 0
  Hello Suppressed    0
```

show mpls rsvp hello graceful restart

- Use to display information about the state of RSVP-TE graceful restart.
- Field descriptions
  - Graceful-restart—State of graceful restart, ON or Off
  - Warning—State of graceful restart attributes
  - Restart time—Graceful restart time, in milliseconds
  - Recovery time—Graceful restart recovery time, in milliseconds
- Example

```
host1#show mpls rsvp hello graceful restart
Graceful restart is ON
    Warning: Graceful restart is NOT active
    Warning: Hellos not configured on all interfaces
    Restart time 60000 milliseconds
    Recovery time 120000 milliseconds
```

show mpls rsvp hello instance

- Use to display summary or detailed information about RSVP-TE hello adjacency instances.
- Field descriptions
  - Peer Address—Address of the peer in the RSVP-TE hello adjacency
  - Interface—Specifier and status of each interface
    - any—Identifies an RSVP-TE node hello peer
  - Interval—Interval at which hellos are sent to the neighbor, in milliseconds
  - Miss Limit—Number of hello messages from the neighbor that can be missed before the adjacency is considered to be down
  - State—State of the adjacency with the neighbor:
    - AdjLost—Adjacency has been lost. Hellos were received from the peer but have timed out. The peer is known to be capable of graceful restart, so the router is waiting for hellos to resume from the peer. The router is actively sending hellos to the peer.

      The router declares the peer to be dead if it does not receive hellos from the peer during the peer's advertised recovery period.

      The router declares the peer to be gracefully restarting if hellos are seen from the peer and its sequence number has changed.

      The router declares the peer to be up if hellos are seen from the peer and its sequence number has not changed.

    - Dead—Hellos were received from the peer but have timed out. The router is not in graceful restart helper mode. The router changes the local hello sequence number and does not send hellos to the peer. The router transitions to Down if new control traffic needs to be sent to the peer or if the peer starts sending control traffic.
    - Down—No hellos have been received from the peer. The router is actively sending hellos to the peer.

    - GR—Graceful restart in progress. The hello sequence number from the peer changed. The router is in graceful restart helper mode and actively sending hellos to the peer.
    - Up—Hellos were received from the peer. The router is actively sending hellos to the peer.
  - Neighbor—IP address of remote hello adjacency peer
  - Local Address—IP address of the local hello adjacency peer
  - Restart Time—Graceful restart time, in milliseconds
  - Recovery Time—Graceful restart recovery time, in milliseconds
  - SrcInstance—Nonzero 32-bit value that represents the sender's hello instance. The value is maintained on a per-neighbor basis. This instance value changes only when the sending peer resets, when the sender's router reboots, or when communication is lost between the hello adjacency peers.
  - DstInstance—32-bit SrcInstance value most recently received from hello adjacency peer; value is zero when no instance has been received from that peer
  - Hellos Sent—Number of hello messages sent
  - Hellos Received—Number of hello messages received
  - Hellos Suppressed—Number of hello messages suppressed; message generation is suppressed when a hello request object is received from the destination node within the hello interval
  - Hellos Acks Sent—Number of acknowledgments sent in response to hello requests received
  - Hellos Acks Received—Number of acknowledgments received in response to hello requests sent
- Example 1

```
host1#show mpls rsvp hello instance
  Up - neighbor is up
  GR - graceful restart is in progress
  Peer
Address    Interface   Interval   Miss Limit  State
--------   ---------   --------   ----------  ------
10.1.1.2   ATM6/1.1    10000      4           Up
10.3.1.2   ATM6/0.3    10000      4           GR
```

- Example 2—Shows that the two peers are identified as RSVP-TE node hellos peers

```
host1#show mpls rsvp hello instance
  Up - neighbor is up
  GR - graceful restart is in progress
  Peer
Address Interface Interval Miss Limit State
-------- --------- -------- ---------- ------
10.1.1.2 <any>    10000 4 Up
10.3.1.2 <any>    10000 4 GR
11.2.3.1 Atm3/1.3 10000 4 GR
```

- Example 3

```
host1#show mpls rsvp hello instance detail
Neighbor 10.1.1.2 on interface ATM6/1.1
  Local Address 10.1.1.1
  Restart Time: 60000 msecs
  Recovery Time: 120000 msecs
  State: Up
  SrcInstance 0x4379F084
  DstInstance 0x4379EA19
  Interval            : 10000
  Miss Limit          : 4
  Hellos Sent         : 0
  Hellos Received     : 382
  Hellos Suppressed   : 381
  Hello Acks Sent     : 382
  Hello Acks Received : 0
```

show mpls tunnels

- Use to display status and configuration for all tunnels or for a specific tunnel in the current router context.
- A result of Incomplete Configuration in the display indicates either no tunnel endpoint or no label distribution protocol.
- Field descriptions
  - Label—Label prepended to packets before being sent across tunnel
  - State—Status of tunnel: Establishing, Traffic Engineering Negotiation, Up, Down, Enabled with Incomplete Config, Disabled with Incomplete Config, Disabled, or Releasing
  - on—Location of tunnel
  - tunnel is announced to—Protocols to which the tunnel is announced
  - metric—Metric type, relative or absolute
  - phb-id—PHB ID supported by this tunnel; for E-LSPs an additional exp-bits entry is displayed after the phb-id entry
  - Mpls Statistics
    - pkts—Number of packets sent across tunnel
    - hcPkts—Number of high-capacity (64-bit) packets sent across tunnel
    - octets—Number of octets sent across tunnel
    - hcOctets—Number of high-capacity (64-bit) octets sent across tunnel
    - errors—Number of packets that are dropped for some reason before being sent
    - discardPkts—Number of packets that are discarded due to lack of buffer space before being sent

- Example

  In the output for Tunnel 2, the line **phb-id 2** indicates that the tunnel is an L-LSP with PHB-ID 2. You can then display the output of the **show mpls phb-id** command to determine the corresponding PSC. For example, the **show mpls phb-id** command described previously in this section indicates that PHB-ID 2 is EF class.

```
host1#show mpls tunnels

Tunnel 1 to   0.0.0.0
  State:  Enabled with Incomplete Config
  tunnel not announced to any IGP
  (Global) Retry forever
       at (Global) interval 5 during Lsp setup if there is route
  (Global) Retry forever
       at (Global) interval 5 during Lsp setup if there is no route
  metric is relative 0

Tunnel 2 to   222.9.1.3
  State:  Up
  Out label 24 on ATM3/0.1 nbr 10.10.11.5
    14 pkts, 0 hcPkts, 2156 octets
    0 hcOctets, 0 errors, 0 discardPkts
  tunnel not announced to any IGP
  (Global) Retry forever
       at (Global) interval 5 during Lsp setup if there is route
  (Global) Retry forever
       at (Global) interval 5 during Lsp setup if there is no route
  metric is relative 0
    phb-id 2
    path option 2
    option is currently used - path is calculated  by isis
        10.10.11.5
        10.10.12.3
        222.9.1.3
    next reoptimization in 1687 seconds
  stacked labels:
FastEthernet2/4.1     222.9.1.3          R0        Out 18 on  tun mpls:1

Tunnel tail-de090106-1-18a for  222.9.1.2
  State:  Up
  In  label 20 on ATM3/0.1
    0 pkts, 0 hcPkts, 0 octets
0 hcOctets, 0 errors, 0 discardPkts
```

show mpls tunnels brief

- Use the **brief** keyword to display a summary of all MPLS tunnels for the current router context or summary information for a specific tunnel in the current router context.
- Field descriptions
  - name/id—Tunnel identifier
  - destination—Tunnel destination; router ID of egress router
  - metric—Value of tunnel metric and whether the metric is relative (R) or absolute (A)
  - state/label/intf—Functional state of tunnel, label for the tunnel, and interface where tunnel resides

- Example

```
host1:pe2#show mpls tunnels brief
name/id          destination  metric  state/label/intf
vpnEgressLabel3  0.0.0.0      R0      Incoming 1048573 on stack
vpnEgressLabel4  0.0.0.0      R0      Incoming 1048572 on stack
pe2-to-pe1       1.1.1.1      R0      Outgoing 300 on atm2/0.60
                 2.2.2.2      R0      Incoming 3000 on atm2/0.70
```