MPLS Configuration Guide for Cisco 8000 Series Routers, IOS XR Release 7.8.x

First Published: 2022-10-12

Last Modified: 2022-10-12

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Preface

This guide describes the Cisco IOS XR configurations.

The preface contains the following sections:

Changes to this Document

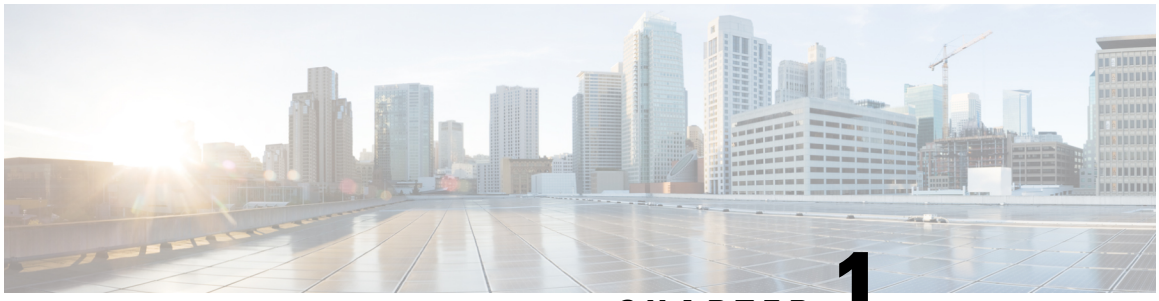
Date	Change Summary
November 2022	Initial release of this document.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.



CHAPTER 1

New and Changed MPLS Features

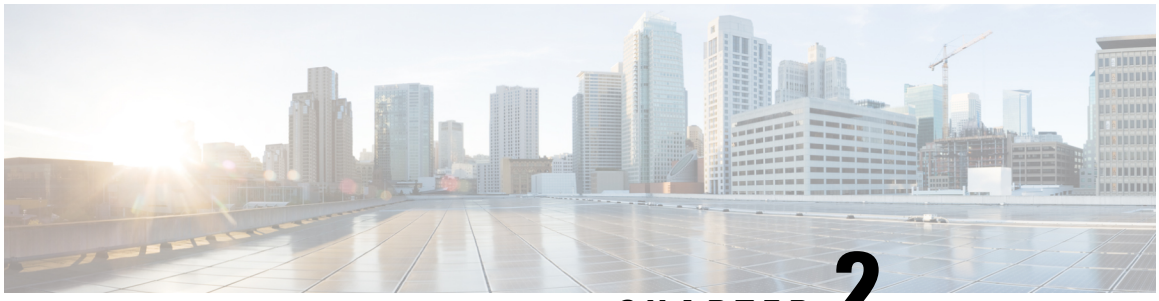
This table summarizes the new and changed feature information for the *MPLS Configuration Guide for Cisco 8000 Series Routers*, and tells you where they are documented.

• New and Changed MPLS Feature Information, on page 1

New and Changed MPLS Feature Information

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
NA	NA	Release 7.8.1	NA



CHAPTER 2

YANG Data Models for MPLS Features

This chapter provides information about the YANG data models for MPLS features.

• Using YANG Data Models, on page 3

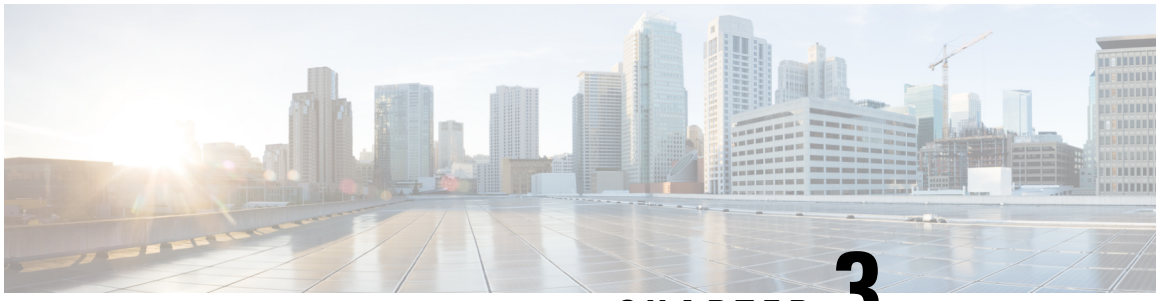
Using YANG Data Models

Cisco IOS XR supports a programmatic way of configuring and collecting operational data of a network device using YANG data models. Although configurations using CLIs are easier and human-readable, automating the configuration using model-driven programmability results in scalability.

The data models are available in the release image, and are also published in the Github repository. Navigate to the release folder of interest to view the list of supported data models and their definitions. Each data model defines a complete and cohesive model, or augments an existing data model with additional XPaths. To view a comprehensive list of the data models supported in a release, navigate to the **Available-Content.md** file in the repository.

You can also view the data model definitions using the YANG Data Models Navigator tool. This GUI-based and easy-to-use tool helps you explore the nuances of the data model and view the dependencies between various containers in the model. You can view the list of models supported across Cisco IOS XR releases and platforms, locate a specific model, view the containers and their respective lists, leaves, and leaf lists presented visually in a tree structure. This visual tree form helps you get insights into nodes that can help you automate your network.

To get started with using the data models, see the *Programmability Configuration Guide*.



CHAPTER 3

Implementing MPLS Label Distribution Protocol

In IP forwarding, when a packet arrives at a router the router looks at the destination address in the IP header, performs a route lookup, and then forwards the packet to the next hop. MPLS is a forwarding mechanism in which packets are forwarded based on labels. Label Distribution Protocols assign, distribute, and install the labels in an MPLS environment. It is the set of procedures and messages by which Label Switched Routers (LSRs) establish LSPs through a network by mapping network-layer routing information directly to data-link layer switched paths. These LSPs may have an endpoint at a directly attached neighbor (comparable to IP hop-by-hop forwarding), or may have an endpoint at a network egress node, enabling switching via all intermediary nodes.

LSPs can be created statically, by RSVP traffic engineering (TE), or by LDP. LSPs created by LDP perform hop-by-hop path setup instead of an end-to-end path. LDP enables LSRs to discover their potential peer routers and to establish LDP sessions with those peers to exchange label binding information. Once label bindings are learned, the LDP is ready to setup the MPLS forwarding plane.

For MPLS LDP, Graceful Out of Resource (OOR) handling is supported from Release 7.3.2 onwards.

Prerequisites for Implementing MPLS Label Distribution Protocol

The following are the prerequisites to implement MPLS LDP:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

- You must be running Cisco IOS XR software.
- You must install a composite mini-image and the MPLS package.

Note This point is not appplicable for a Cisco NCS 540 Series Router.

- You must activate IGP.
- We recommend to use a lower session holdtime bandwidth such as neighbors so that a session down occurs before an adjacency-down on a neighbor. Therefore, the following default values for the hello times are listed:
  - Holdtime is 15 seconds.
  - Interval is 5 seconds.

  For example, the LDP session holdtime can be configured as 30 seconds by using the **holdtime** command.

Restrictions and Recommendations

The following restrictions and recommendations apply to the MPLS LDP CSC feature:

- Only IPv4 address family is supported for a default or a non-default VRF.
- No T-LDP support in a VRF context.
- An address family under VRF and VRF interface must be configured for non-default VRFs.
- Following scenarios are not supported :
  - Different VRFs between a given PE-CE device pair (VRFs configured on different links and interfaces)
  - LDP/BGP CSC co-existence on a given VRF between a given PE-CE device pair:
    - Single link
    - Parallel links: LDP CSC on one link and BGP CSC on the other
- LDP router-id must be configured per-VRF. If not configured for non-default VRF, LDP computes router-id from available loopback interfaces under the VRF.
- It is recommended to configure a routable discovery transport address under a VRF IPv4 address-family submode for deterministic transport endpoint and connection.
- When LDP CSC is configured and in use:
  - BGP label allocation policy for VRF prefixes must be per-prefix
  - Selective VRF Download (SVD) feature must be disabled

Information About Implementing Cisco MPLS LDP

To implement MPLS LDP, you should understand these concepts:

IP LDP Fast Reroute Loop Free Alternate

The IP Fast Reroute is a mechanism that enables a router to rapidly switch traffic, after an adjacent link failure, node failure, or both, towards a pre-programmed loop-free alternative (LFA) path. This LFA path is used to switch traffic until the router installs a new primary next hop again, as computed for the changed network topology.

The goal of LFA FRR is to reduce failure reaction time to 50 milliseconds by using a pre-computed alternate next hop, in the event that the currently selected primary next hop fails, so that the alternate can be rapidly used when the failure is detected.

This feature targets to address the fast convergence ability by detecting, computing, updating or enabling prefix independent pre-computed alternate loop-free paths at the time of failure.

IGP pre-computes a backup path per IGP prefix. IGP selects one and only one backup path per primary path. RIB installs the best path and download path protection information to FIB by providing correct annotation for protected and protecting paths. FIB pre-installs the backup path in dataplane. Upon the link or node failure, the routing protocol detects the failure, all the backup paths of the impacted prefixes are enabled in a prefix-independent manner.

Prerequisites

The Label Distribution Protocol (LDP) can use the loop-free alternates as long as these prerequisites are met:

The Label Switching Router (LSR) running LDP must distribute its labels for the Forwarding Equivalence Classes (FECs) it can provide to all its neighbors, regardless of whether they are upstream, or not.

There are two approaches in computing LFAs:

- **Link-based (per-link)**--In link-based LFAs, all prefixes reachable through the primary (protected) link share the same backup information. This means that the whole set of prefixes, sharing the same primary, also share the repair or fast reroute (FRR) ability. The per-link approach protects only the next hop address. The per-link approach is suboptimal and not the best for capacity planning. This is because all traffic is redirected to the next hop instead of being spread over multiple paths, which may lead to potential congestion on link to the next hop. The per-link approach does not provide support for node protection.
- **Prefix-based (per-prefix)**--Prefix-based LFAs allow computing backup information per prefix. It protects the destination address. The per-prefix approach is the preferred approach due to its greater applicability, and the greater protection and better bandwidth utilization that it offers.

Note The repair or backup information computed for a given prefix using prefix-based LFA may be different from the computed by link-based LFA.

The per-prefix LFA approach is preferred for LDP IP Fast Reroute LFA for these reasons:

- Better node failure resistance

- Better capacity planning and coverage

Features Not Supported

These interfaces and features are not supported for the IP LDP Fast Reroute Loop Free Alternate feature:

- BVI interface (IRB) is not supported either as primary or backup path.
- GRE tunnel is not supported either as primary or backup path.
- In a multi-topology scenerio, the route in topology T can only use LFA within topology T. Hence, the availability of a backup path depends on the topology.

For more information about configuring the IP Fast Reroute Loop-free alternate , see Implementing IS-IS on Cisco IOS XR Software module of the *Routing Configuration Guide for Cisco 8000 Series Routers*.

IS-IS

Intermediate System-to-Intermediate System (IS-IS) is an Interior Gateway Protocol (IGP) that advertises link-state information throughout the network to create a picture of the network topology. IPv6 IS-IS extends the address families supported by IS-IS to include IPv6, in addition to IPv4.

Previously, IS-IS supported registration of only LDP IPv4 sync status change. This has now been enhanced to support registration of notifications of LDP IPv6 sync status change. IS-IS determines the link-metrics to be advertised based on the LDP-IGP sync status on the IPv4 and IPv6 address families.

IS-IS supports non-stop forwarding (NSF) by preserving the LDPv6-IGP sync status across high availability (HA) events of IS-IS process restarts and failover.

IS-IS also supports LDPv6-IGP sync for LFA-FRR by checking the sync status of the backup interface (if it is configured with LDP IPv6 sync).

Label Acceptance Control (Inbound Filtering)

By default, LDP accepts labels (as remote bindings) for all prefixes from all peers. LDP operates in liberal label retention mode, which instructs LDP to keep remote bindings from all peers for a given prefix. For security reasons, or to conserve memory, you can override this behavior by configuring label binding acceptance for set of prefixes from a given peer.

The ability to filter remote bindings for a defined set of prefixes is also referred to as *LDP inbound label filtering*.

Note Inbound filtering can also be implemented using an outbound filtering policy; however, you may not be able to implement this system if an LDP peer resides under a different administration domain. When both inbound and outbound filtering options are available, we recommend that you use outbound label filtering.

Label Advertisement Control (Outbound Filtering)

By default, LDP advertises labels for all the prefixes to all its neighbors. When this is not desirable (for scalability and security reasons), you can configure LDP to perform outbound filtering for local label

advertisement for one or more prefixes to one more peers. This feature is known as *LDP outbound label filtering*, or *local label advertisement control*.

Label Switched Paths

LSPs are created in the network through MPLS. They can be created statically, by RSVP traffic engineering (TE), or by LDP. LSPs created by LDP perform hop-by-hop path setup instead of an end-to-end path.

LDP Control Plane

The control plane enables label switched routers (LSRs) to discover their potential peer routers and to establish LDP sessions with those peers to exchange label binding information.

LDP Control Plane: Bindings Advertisement

LDP base specification allows exchange of IPv4/IPv6 bindings (address/label) on an established session. When both IPv4 and IPv6 address families are enabled under LDP, LDP distributes address/label bindings for both address families to its established peer according to local policies. Following are a few significant points pertaining to bindings support for IPv6:

- LDP allocates/advertises local label bindings for link-local IPv6 address prefixes. If received, such FEC bindings are ignored.
- LDP sends only the Prefix FEC of the single address family type in a FEC TLV and not include both. If such a FEC binding is received, the entire message is ignored.
- LDP sends only the addresses belonging to same address family in a single address list TLV (in address or address withdraw message).

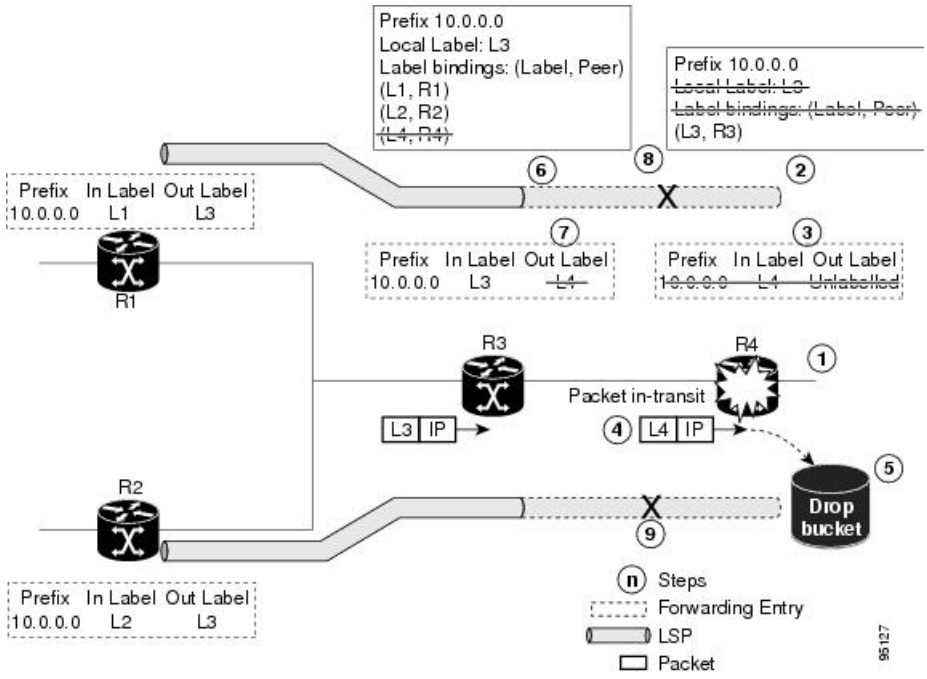
If an address family is not enabled on receiving LSR, LDP discards any bindings received from peer(s) for the address family. This means that when address family is enabled, LDP needs to reset existing sessions with the peers in order to re-learn the discarded bindings. The implementation is optimized to reset only those sessions which were previously known to be dual-stack and had sent bindings for both address families.

Control Plane Failure

When a control plane failure occurs, connectivity can be affected. The forwarding states installed by the router control planes are lost, and the in-transit packets could be dropped, thus breaking NSF.

Figure 1: Control Plane Failure

This figure illustrates a control plane failure and shows the process and results of a control plane failure leading to loss of connectivity.



1. The R4 LSR control plane restarts.
2. LIB is lost when the control plane restarts.
3. The forwarding states installed by the R4 LDP control plane are immediately deleted.
4. Any in-transit packets flowing from R3 to R4 (still labeled with L4) arrive at R4.
5. The MPLS forwarding plane at R4 performs a lookup on local label L4 which fails. Because of this failure, the packet is dropped and NSF is not met.
6. The R3 LDP peer detects the failure of the control plane channel and deletes its label bindings from R4.
7. The R3 control plane stops using outgoing labels from R4 and deletes the corresponding forwarding state (rewrites), which in turn causes forwarding disruption.
8. The established LSPs connected to R4 are terminated at R3, resulting in broken end-to-end LSPs from R1 to R4.
9. The established LSPs connected to R4 are terminated at R3, resulting in broken LSPs end-to-end from R2 to R4.

Default Transport Address

LDP computes default local transport address for IPv6 from its IPv6 interface or address database by picking the lowest operational loopback interface with global unicast IPv6 address. This means that any change in this loopback state or address, flaps or changes the default transport address for IPv6 and may cause session flaps using such an address as transport endpoint. For example, if a session is currently active on Loopback2 as during it's inception it was the lowest loopback with an IPv6 address, and a lower loopback, Loopback0, is configured with an IPv6 address, the session does not flap. However, if it does flap, the next time the session is attempted, Loopback0 is used.

The session flaps when configuring discovery transport address explicitly.

Use the `discovery transport-address` command under the LDP address family submode to specify the global transport address for IPv4 or IPv6.

It is recommended to configure global transport-address for IPv6 address family to avoid a potentially unstable default transport address.

Label Distribution Protocol Discovery Parameters

Discovery parameter specifies the time periods between transmitted and not received hello messages.

Configuration Example

A discovery parameter specifies time of the discovered neighbor (15 seconds) which is kept without receipt of any subsequent hello messages. After the specified time period, there is an interval of 5 seconds between the transmission of consecutive hello messages.

Configuration of Label Distribution Protocol Discovery Parameters

```
Router(config)#mpls ldp
Router(config-ldp)#router-id 192.168.70.1
Router(config-ldp)#discovery hello holdtime 15
Router(config-ldp)#discovery targeted-hello holdtime 5
Router(config-ldp)#commit
```

Verification

Displays all the current MPLS LDP parameters.

```
RP/0/RP0/CPU0:router# show mpls ldp parameters
LDP Parameters:
Role: Active
Protocol Version: 1
Router ID: 192.168.70.1


Discovery:
Link Hellos:     Holdtime:15 sec, Interval:5 sec
Targeted Hellos: Holdtime:5 sec, Interval:10 sec
Quick-start: Enabled (by default)
Transport address:       IPv4: 192.168.70.1
```

Downstream on Demand

The Downstream on demand feature adds support for downstream-on-demand mode, where the label is not advertised to a peer, unless the peer explicitly requests it. At the same time, since the peer does not automatically advertise labels, the label request is sent whenever the next-hop points out to a peer that no remote label has been assigned.

To enable downstream-on-demand mode, this configuration must be applied at mpls ldp configuration mode:

mpls ldp downstream-on-demand with *ACL*

The ACL contains a list of peer IDs that are configured for downstream-on-demand mode. When the ACL is changed or configured, the list of established neighbors is traversed. If a session's downstream-on-demand configuration has changed, the session is reset in order that the new down-stream-on-demand mode can be configured. The reason for resetting the session is to ensure that the labels are properly advertised between the peers. When a new session is established, the ACL is verified to determine whether the session should negotiate for downstream-on-demand mode. If the ACL does not exist or is empty, downstream-on-demand mode is not configured for any neighbor.

For it to be enabled, the Downstream on demand feature has to be configured on both peers of the session. If only one peer in the session has downstream-on-demand feature configured, then the session does not use downstream-on-demand mode.

If, after, a label request is sent, and no remote label is received from the peer, the router will periodically resend the label request. After the peer advertises a label after receiving the label request, it will automatically readvertise the label if any label attribute changes subsequently.

Explicit-Null and Implicit-Null Labels

Cisco MPLS LDP uses null label, implicit or explicit, as local label for routes or prefixes that terminate on the given LSR. These routes include all local, connected, and attached networks. By default, the null label is **implicit-null** that allows LDP control plane to implement penultimate hop popping (PHOP) mechanism. When this is not desirable, you can configure **explicit-null** that allows LDP control plane to implement ultimate hop popping (UHOP) mechanism. You can configure this explicit-null feature on the ultimate hop LSR. This configuration knob includes an access-list to specify the IP prefixes for which PHOP is desired.

This new enhancement allows you to configure implicit-null local label for **non-egress (ultimate hop LSR)** prefixes by using the **implicit-null-override** command. This enforces implicit-null local label for a specific prefix even if the prefix requires a non-null label to be allocated by default. For example, by default, an LSR allocates and advertises a non-null label for an IGP route. If you wish to terminate LSP for this route on penultimate hop of the LSR, you can enforce implicit-null label allocation and advertisement for this prefix using **implicit-null-override** feature.

Note If a given prefix is permitted in both explicit-null and implicit-null-override feature, then implicit-null-override supercedes and an implicit-null label is allocated and advertised for the prefix.

In order to enable implicit-null-override mode, this configuration must be applied at MPLS LDP label configuration mode:

```
mpls ldp
   label
     implicit-null-override for <prefix><ACL>

  !
```

This feature works with any prefix including static, IGP, and BGP, when specified in the ACL.

Label Distribution Protocol Interior Gateway Protocol Synchronization

Lack of synchronization between LDP and Interior Gateway Protocol (IGP) can cause MPLS traffic loss. Upon link up, for example, IGP can advertise and use a link before LDP convergence has occurred or, a link may continue to be used in IGP after an LDP session goes down.

LDP IGP synchronization coordinates LDP and IGP so that IGP advertises links with regular metrics only when MPLS LDP is converged on that link. LDP considers a link converged when at least one LDP session is up and running on the link for which LDP has sent its applicable label bindings and received at least one label binding from the peer. LDP communicates this information to IGP upon link up or session down events and IGP acts accordingly, depending on sync state.

In the event, an LDP graceful restart session disconnect, a session is treated as converged as long as the graceful restart neighbor is timed out. Additionally, upon local LDP restart, a check-point recovered LDP graceful restart session is used and treated as converged and is given an opportunity to connect and resynchronize.

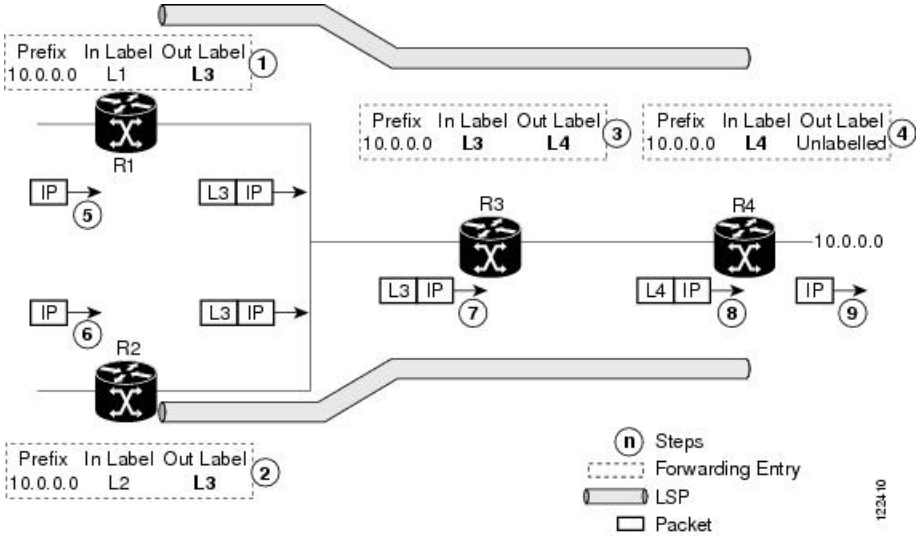
Under certain circumstances, it might be required to delay declaration of re-synchronization to a configurable interval. LDP provides a configuration option to delay declaring synchronization up for up to 60 seconds. LDP communicates this information to IGP upon linkup or session down events.

LDP Forwarding

Once label bindings are learned, the LDP control plane is ready to setup the MPLS forwarding plane as shown in the following figure.

Figure 2: Forwarding Setup

Once label bindings are learned, the LDP control plane is ready to setup the MPLS forwarding plane as shown in this figure.



1. Because R3 is next hop for 10.0.0.0 as notified by the FIB, R1 selects label binding from R3 and installs forwarding entry (Layer 1, Layer 3).
2. Because R3 is next hop for 10.0.0.0 (as notified by FIB), R2 selects label binding from R3 and installs forwarding entry (Layer 2, Layer 3).
3. Because R4 is next hop for 10.0.0.0 (as notified by FIB), R3 selects label binding from R4 and installs forwarding entry (Layer 3, Layer 4).
4. Because next hop for 10.0.0.0 (as notified by FIB) is beyond R4, R4 uses NO-LABEL as the outbound and installs the forwarding entry (Layer 4); the outbound packet is forwarded IP-only.

5. Incoming IP traffic on ingress LSR R1 gets label-imposed and is forwarded as an MPLS packet with label L3.
6. Incoming IP traffic on ingress LSR R2 gets label-imposed and is forwarded as an MPLS packet with label L3.
7. R3 receives an MPLS packet with label L3, looks up in the MPLS label forwarding table and switches this packet as an MPLS packet with label L4.
8. R4 receives an MPLS packet with label L4, looks up in the MPLS label forwarding table and finds that it should be Unlabeled, pops the top label, and passes it to the IP forwarding plane.
9. IP forwarding takes over and forwards the packet onward.

Note For local labels, only up to 12000 rewrites are supported. If the rewrites exceed this limit, MPLS LSD or MPLS LDP or both the processes may crash.

Setting up Label Distribution Protocol Graceful Restart

Configuration Example

This example shows how to configure LDP graceful restart. In this example, the amount of time that a neighboring router maintains the forwarding state about the gracefully restarting router is specified as 180 seconds. Also, the amount of time the LDP neighbor should wait for a reconnection from the gracefully restarting router in the event of a LDP session failure is specified as 169 seconds.

```
Router(config)#mpls ldp
Router(config-ldp)#interface TenGigE 0/0/0/5
Router(config-ldp-if)#exit
Router(config-ldp)#graceful-restart
Router(config-ldp)#graceful-restart forwarding-state-holdtime 180
Router(config-ldp)#graceful-restart reconnect-timeout 169
Router(config-ldp)#commit
```

Verification

```
RP/0/RP0/CPU0:router#show mpls ldp graceful-restart
Forwarding State Hold timer : Not Running
GR Neighbors : 1

Neighbor ID      Up  Connect Count  Liveness Timer      Recovery Timer
---------------  --  -------------  ------------------  ------------------
8.8.8.8          Y        1               -                   -
```

```
RP/0/RP0/CPU0:router#show mpls ldp parameters
Graceful Restart:Enabled
Reconnect Timeout:169 sec, Forwarding State Holdtime:180 sec
NSR: Disabled, Not Sync-ed
```

Phases in Graceful Restart

The graceful restart mechanism is divided into different phases:

Control communication failure detection

Control communication failure is detected when the system detects either:

- Missed LDP hello discovery messages
- Missed LDP keepalive protocol messages
- Detection of Transmission Control Protocol (TCP) disconnection a with a peer

Forwarding state maintenance during failure

Persistent forwarding states at each LSR are achieved through persistent storage (checkpoint) by the LDP control plane. While the control plane is in the process of recovering, the forwarding plane keeps the forwarding states, but marks them as stale. Similarly, the peer control plane also keeps (and marks as stale) the installed forwarding rewrites associated with the node that is restarting. The combination of local node forwarding and remote node forwarding plane states ensures NSF and no disruption in the traffic.

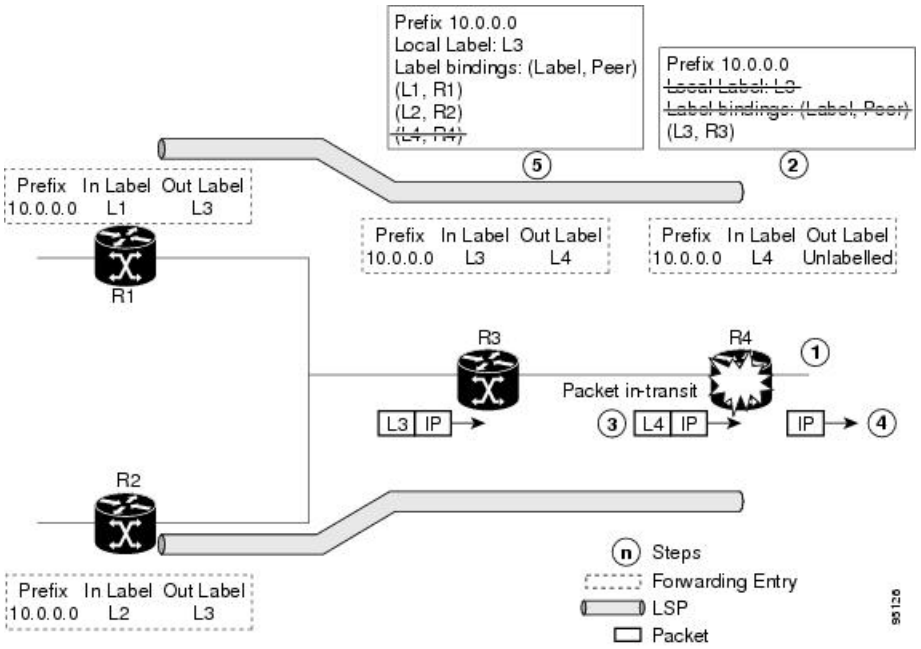
Control state recovery

Recovery occurs when the session is reestablished and label bindings are exchanged again. This process allows the peer nodes to synchronize and to refresh stale forwarding states.

Recovery with Graceful-Restart

Figure 3: Recovering with Graceful Restart

This figure illustrates the process of failure recovery using graceful restart.



1. The router R4 LSR control plane restarts.

2. With the control plane restart, LIB is gone but forwarding states installed by R4's LDP control plane are not immediately deleted but are marked as stale.
3. Any in-transit packets from R3 to R4 (still labeled with L4) arrive at R4.
4. The MPLS forwarding plane at R4 performs a successful lookup for the local label L4 as forwarding is still intact. The packet is forwarded accordingly.
5. The router R3 LDP peer detects the failure of the control plane and channel and deletes the label bindings from R4. The peer, however, does not delete the corresponding forwarding states but marks them as stale.
6. At this point there are no forwarding disruptions.
7. The peer also starts the neighbor reconnect timer using the reconnect time value.
8. The established LSPs going toward the router R4 are still intact, and there are no broken LSPs.

When the LDP control plane recovers, the restarting LSR starts its forwarding state hold timer and restores its forwarding state from the checkpointed data. This action reinstates the forwarding state and entries and marks them as old.

The restarting LSR reconnects to its peer, indicated in the FT Session TLV, that it either was or was not able to restore its state successfully. If it was able to restore the state, the bindings are resynchronized.

The peer LSR stops the neighbor reconnect timer (started by the restarting LSR), when the restarting peer connects and starts the neighbor recovery timer. The peer LSR checks the FT Session TLV if the restarting peer was able to restore its state successfully. It reinstates the corresponding forwarding state entries and receives binding from the restarting peer. When the recovery timer expires, any forwarding state that is still marked as stale is deleted.

If the restarting LSR fails to recover (restart), the restarting LSR forwarding state and entries will eventually timeout and is deleted, while neighbor-related forwarding states or entries are removed by the Peer LSR on expiration of the reconnect or recovery timers.

LDP Nonstop Routing

LDP nonstop routing (NSR) functionality makes failures, such as Route Processor (RP) or Distributed Route Processor (DRP) failover, invisible to routing peers with minimal to no disruption of convergence performance. By default, NSR is globally enabled on all LDP sessions except AToM.

A disruption in service may include any of these events:

- Route processor (RP) or distributed route processor (DRP) failover
- LDP process restart
- In-service system upgrade (ISSU)
- Minimum disruption restart (MDR)

Note Unlike graceful restart functionality, LDP NSR does not require protocol extensions and does not force software upgrades on other routers in the network, nor does LDP NSR require peer routers to support NSR.

Process failures of active TCP or LDP results in session loss and, as a result, NSR cannot be provided unless RP switchover is configured as a recovery action. For more information about how to configure switchover as a recovery action for NSR, see *Configuring Transports* module in *IP Addresses and Services Configuration Guide for Cisco 8000 Series Routers*.

Local Label Allocation Control

Label Distribution Protocol allocates local labels for all prefixes that are not Border Gateway Protocol (BGP) prefixes[1]. This is acceptable when LDP is used for applications other than Layer 3 virtual private networks (L3VPN) core transport. When LDP is used to set up transport LSPs for L3VPN traffic in the core, it is not efficient or even necessary to allocate and advertise local labels for, potentially, thousands of IGP prefixes. In such a case, LDP is typically required to allocate and advertise local label for loopback /32 addresses for PE routers. This is accomplished using LDP local label allocation control, where an access list can be used to limit allocation of local labels to a set of prefixes. Limiting local label allocation provides several benefits, including reduced memory usage requirements, fewer local forwarding updates, and fewer network and peer updates.

Redistributing MPLS LDP Routes into BGP

Perform this task to redistribute Border Gateway Protocol (BGP) autonomous system into an MPLS LDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **redistribute bgp**
4. **end** or **commit**
5. **show run mpls ldp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**redistribute bgp** **Example:**	Allows the redistribution of BGP routes into an MPLS LDP processes.

[1] For L3VPN Inter-AS option C, LDP may also be required to assign local labels for some BGP prefixes.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-ldp)# redistribute bgp advertise-to acl_1`	**Note** Autonomous system numbers (ASNs) are globally unique identifiers used to identify autonomous systems (ASs) and enable ASs to exchange exterior routing information between neighboring ASs. A unique ASN is allocated to each AS for use in BGP routing. ASNs are encoded as 2-byte numbers and 4-byte numbers in BGP.
Step 4	**end** or **commit**	• When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 5	**show run mpls ldp** **Example:** `RP/0/RP0/CPU0:router# show run mpls ldp`	Displays information about the redistributed route information.

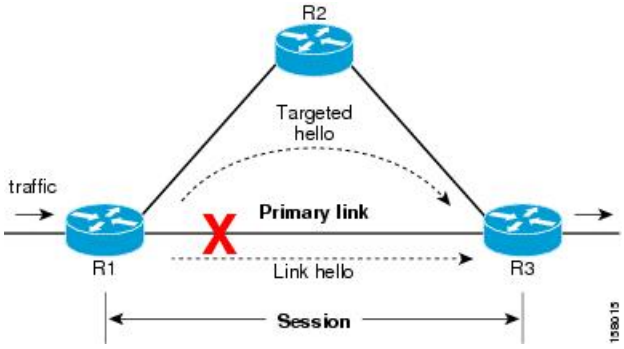
Session Protection

When a link comes up, IP converges earlier and much faster than MPLS LDP and may result in MPLS traffic loss until MPLS convergence. If a link flaps, the LDP session will also flap due to loss of link discovery. LDP session protection minimizes traffic loss, provides faster convergence, and protects existing LDP (link) sessions by means of "parallel" source of targeted discovery hello. An LDP session is kept alive and neighbor label bindings are maintained when links are down. Upon reestablishment of primary link adjacencies, MPLS convergence is expedited as LDP need not relearn the neighbor label bindings.

LDP session protection lets you configure LDP to automatically protect sessions with all or a given set of peers (as specified by peer-acl). When configured, LDP initiates backup targeted hellos automatically for neighbors for which primary link adjacencies already exist. These backup targeted hellos maintain LDP sessions when primary link adjacencies go down.

The Session Protection figure illustrates LDP session protection between neighbors R1 and R3. The primary link adjacency between R1 and R3 is directly connected link and the backup; targeted adjacency is maintained between R1 and R3. If the direct link fails, LDP link adjacency is destroyed, but the session is kept up and running using targeted hello adjacency (through R2). When the direct link comes back up, there is no change in the LDP session state and LDP can converge quickly and begin forwarding MPLS traffic.

Figure 4: Session Protection



Note When LDP session protection is activated (upon link failure), protection is maintained for an unlimited period time.

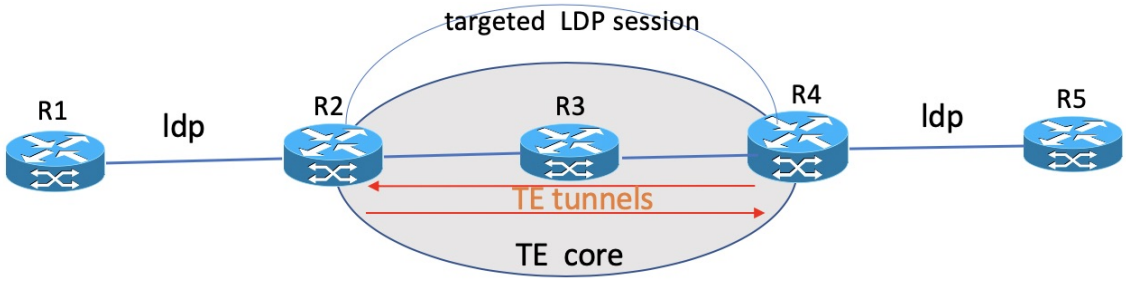
LDP Over RSVP LSR Support

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
LDP Over RSVP LSR Support	Release 7.3.1	With this feature, users can transport LDP traffic over an RSVP-TE network automatically, through a targeted LDP session. The automatic configuration for LDP over RSVP-TE supports 1000 TE tunnels.

Consider this topology of an RSVP-TE network spanning R2 to R4. LDP traffic is transported from R1 to R5. A targeted LDP session is established between R2 and R4 so that LDP traffic is transported over the TE tunnel network.

Figure 5: LDP Over RSVP



No additional configuration is required to enable the LDP over RSVP-TE function. Up to 1000 tunnels are supported by default. The **autoroute announce** command is enabled on the edge routers of the RSVP-TE network.

If you need more than 1000 TE tunnels, enable the **hw-module profile cef te-tunnel highscale-no-ldp-over-te** command on the edge routers R2 and R4. However, when you enable this command, the LDP over TE feature gets disabled.

The following configuration disables the LDP over TE function, and allows you run more than 1000 TE tunnels.

```
Router# configure terminal
Router(config)# hw-module profile cef te-tunnel highscale-no-ldp-over-te
Router(config)# commit
Router# reload
```

How to Implement MPLS LDP

A typical MPLS LDP deployment requires coordination among several global neighbor routers. Various configuration tasks are required to implement MPLS LDP :

Implementing MPLS Label Distribution Protocol

MPLS (Multi Protocol Label Switching) is a forwarding mechanism based on label switching. In an MPLS network, data packets are assigned labels and packet-forwarding decisions are taken based on the contents of the label. To switch labeled packets across the MPLS network, predetermined paths are established for various source-destination pairs. These predetermined paths are known as Label Switched Paths (LSPs). To establish LSPs, MPLS signaling protocols are used. Label Distribution Protocol (LDP) is an MPLS signaling protocol used for establishing LSPs. This module provides information about how to configure MPLS LDP.

Enabling MLDP

Perform this task to enable Multicast Label Distribution Protocol (MLDP) in MPLS LDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**

4. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**mpls ldp** **Example:** `Router(config)# mpls ldp`	Enters the MPLS LDP configuration mode.

Step 3	**mldp**
	Example:
	`Router(config-ldp)# mldp`
	`Router(config-ldp-mldp)#`
		Enables MLDP.

Step 4	**end** or **commit**
	Example:
	`Router(config-ldp-mldp)# end`
	or
	`Router(config-ldp-mldp)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP Make-Before-Break

Perform this task to enable the make-before-break (MBB) feature in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**
5. **make-before-break** [**delay** *seconds*]
6. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**make-before-break** [**delay** *seconds*] **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp-af)# make-before-break delay 10`	Enables the make-before-break feature. (Optional) Configures the MBB forwarding delay in seconds. Range is 0 to 600.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)?`
		`[cancel]:`

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# commit`	• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP MoFRR

Perform this task to enable multicast only fast reroute (MoFRR) support in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**
5. **mofrr**
6. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:**	Enables MLDP.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-ldp)# mldp`	
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**mofrr** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp-af)# mofrr`	Enables MoFRR support.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP Recursive FEC

Perform this task to enable recursive forwarding equivalence class (FEC) support in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**

5. **recursive-fec**
6. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**recursive-fec** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp-af)# recursive-fec`	Enables recursive FEC support.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP Static Multipoint to Multipoint LSP

Perform this task to enable static multipoint to multipoint (MP2MP) LSP support in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**
5. **static mp2mp** *ip-address*
6. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)#`	Enables MLDP for IPv4 address family.

	Command or Action	Purpose
	`address-family ipv4`	
Step 5	**static mp2mp** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp-af)# static mp2mp 10.10.10.10 1`	Enables static MP2MP LSP support and specifies MP2MP LSP root IP address followed by the number of LSPs in the range 1 to 1000.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP Static Point to Multipoint LSP

Perform this task to enable static point to multipoint (P2MP) LSP support in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**
5. **static p2mp** *ip-address*
6. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**static p2mp** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp-af)# static p2mp 10.0.0.1 1`	Enables static P2MP LSP support and specifies P2MP LSP root IP address followed by the number of LSPs in the range 1 to 1000.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

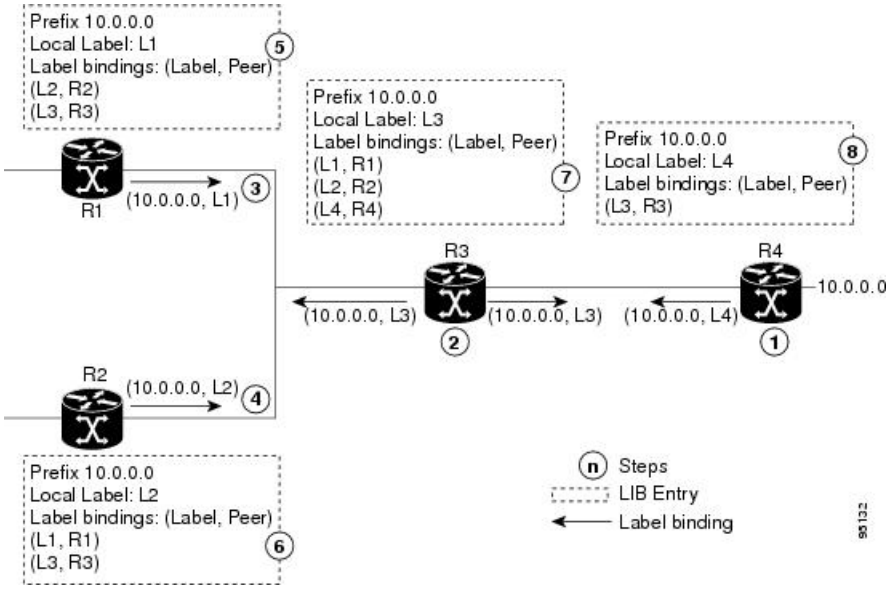
	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Exchanging Label Bindings

LDP creates LSPs to perform the hop-by-hop path setup so that MPLS packets can be transferred between the nodes on the MPLS network.

Figure 6: Setting Up Label Switched Paths

This figure illustrates the process of label binding exchange for setting up LSPs.



For a given network (10.0.0.0), hop-by-hop LSPs are set up between each of the adjacent routers (or, nodes) and each node allocates a local label and passes it to its neighbor as a binding:

1. R4 allocates local label L4 for prefix 10.0.0.0 and advertises it to its neighbors (R3).
2. R3 allocates local label L3 for prefix 10.0.0.0 and advertises it to its neighbors (R1, R2, R4).
3. R1 allocates local label L1 for prefix 10.0.0.0 and advertises it to its neighbors (R2, R3).
4. R2 allocates local label L2 for prefix 10.0.0.0 and advertises it to its neighbors (R1, R3).
5. R1's label information base (LIB) keeps local and remote labels bindings from its neighbors.
6. R2's LIB keeps local and remote labels bindings from its neighbors.
7. R3's LIB keeps local and remote labels bindings from its neighbors.

8. R4's LIB keeps local and remote labels bindings from its neighbors.

Configuring Label Advertisement Control (Outbound Filtering)

Perform this task to configure label advertisement (outbound filtering).

By default, a label switched router (LSR) advertises all incoming label prefixes to each neighboring router. You can control the exchange of label binding information using the **mpls ldp label advertise** command. Using the optional keywords, you can advertise selective prefixes to all neighbors, advertise selective prefixes to defined neighbors, or disable label advertisement to all peers for all prefixes.

Note Prefixes and peers advertised selectively are defined in the access list.

Before you begin

Before configuring label advertisement, enable LDP and configure an access list.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **label advertise** { **disable** | **for** *prefix-acl* [**to** *peer-acl*] | **interface** *type interface-path-id* }
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.

Step 3	**label advertise** { **disable**	**for** *prefix-acl* [**to** *peer-acl*]	**interface** *type interface-path-id* }
			Example:
			`RP/0/RP0/CPU0:router(config-ldp)# label advertise interface POS 0/1/0/0`
			`RP/0/RP0/CPU0:router(config-ldp)# for pfx_acl1 to peer_acl1`
				Configures label advertisement by specifying one of the following options:
				disable
				Disables label advertisement to all peers for all prefixes (if there are no other conflicting rules).
				interface
				Specifies an interface for label advertisement of an interface address.

	Command or Action	Purpose
		for *prefix-acl* **to** *peer-acl* Specifies neighbors to advertise and receive label advertisements.
Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Setting Up Implicit-Null-Override Label

Perform this task to configure implicit-null label for non-egress prefixes.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **label**
4. **implicit-null-override for** *access-list*
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**label** **Example:** `RP/0/RP0/CPU0:router(config-ldp-af)# label`	Configures the allocation, advertisement ,and acceptance of labels.

	Command or Action	Purpose
Step 4	**implicit-null-override for** *access-list* **Example:** `RP/0/RP0/CPU0:router(config-ldp-af-lbl)# implicit-null-override for 70`	Configures implicit-null local label for non-egress prefixes. **Note** This feature works with any prefix including static, IGP, and BGP, when specified in the ACL.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Setting Up LDP Forwarding

Perform this task to set up LDP forwarding.

By default, the LDP control plane implements the penultimate hop popping (PHOP) mechanism. The PHOP mechanism requires that label switched routers use the implicit-null label as a local label for the given Forwarding Equivalence Class (FEC) for which LSR is the penultimate hop. Although PHOP has certain advantages, it may be required to extend LSP up to the ultimate hop under certain circumstances (for example, to propagate MPL QoS). This is done using a special local label (explicit-null) advertised to the peers after which the peers use this label when forwarding traffic toward the ultimate hop (egress LSR).

Before you begin

Stable router ID is required at either end of the link to ensure the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system will default to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **explicit-null**
4. Use the **commit** or **end** command.
5. (Optional) **show mpls ldp forwarding**
6. (Optional) **show mpls forwarding**
7. (Optional) **ping** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**explicit-null** **Example:** `RP/0/RP0/CPU0:router(config-ldp-af)# explicit-null`	Causes a router to advertise an explicit null label in situations where it normally advertises an implicit null label (for example, to enable an ultimate-hop disposition instead of PHOP).
Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 5	(Optional) **show mpls ldp forwarding** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp forwarding`	Displays the MPLS LDP view of installed forwarding states (rewrites). **Note** For local labels, only up to 12000 rewrites are supported. If the rewrites exceed this limit, MPLS LSD or MPLS LDP or both the processes may crash.
Step 6	(Optional) **show mpls forwarding** **Example:** `RP/0/RP0/CPU0:router# show mpls forwarding`	Displays a global view of all MPLS installed forwarding states (rewrites) by various applications (LDP, TE, and static).
Step 7	(Optional) **ping** *ip-address* **Example:** `RP/0/RP0/CPU0:router# ping 192.168.2.55`	Checks for connectivity to a particular IP address (going through MPLS LSP as shown in the **show mpls forwarding** command).

Setting Up LDP Neighbors

Perform this task to set up LDP neighbors.

Before you begin

Stable router ID is required at either end of the link to ensure the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system will default to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **interface** *type interface-path-id*
4. **discovery transport-address** [*ip-address* | **interface**]
5. **exit**
6. **holdtime** *seconds*
7. **neighbor** *ip-address* **password** [*encryption*] *password*
8. **backoff** *initial maximum*
9. Use the **commit** or **end** command.
10. (Optional) **show mpls ldp neighbor**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface POS 0/1/0/0`	Enters interface configuration mode for the LDP protocol.
Step 4	**discovery transport-address** [*ip-address* | **interface**] **Example:** or `RP/0/RP0/CPU0:router(config-ldp-if-af)# discovery transport-address interface`	Provides an alternative transport address for a TCP connection. • Default transport address advertised by an LSR (for TCP connections) to its peer is the router ID. • Transport address configuration is applied for a given LDP-enabled interface.

	Command or Action	Purpose
		• If the interface version of the command is used, the configured IP address of the interface is passed to its neighbors as the transport address.
Step 5	**exit** **Example:** `RP/0/RP0/CPU0:router(config-ldp-if)# exit`	Exits the current configuration mode.
Step 6	**holdtime** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# holdtime 30`	Changes the time for which an LDP session is maintained in the absence of LDP messages from the peer. • Outgoing keepalive interval is adjusted accordingly (to make three keepalives in a given holdtime) with a change in session holdtime value. • Session holdtime is also exchanged when the session is established. • In this example holdtime is set to 30 seconds, which causes the peer session to timeout in 30 seconds, as well as transmitting outgoing keepalive messages toward the peer every 10 seconds.
Step 7	**neighbor** *ip-address* **password** [*encryption*] *password* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# neighbor 192.168.2.44 password secretpasswd`	Configures password authentication (using the TCP MD5 option) for a given neighbor.
Step 8	**backoff** *initial maximum* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# backoff 10 20`	Configures the parameters for the LDP backoff mechanism. The LDP backoff mechanism prevents two incompatibly configured LSRs from engaging in an unthrottled sequence of session setup failures. If a session setup attempt fails due to such incompatibility, each LSR delays its next attempt (backs off), increasing the delay exponentially with each successive failure until the maximum backoff delay is reached.
Step 9	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

	Command or Action	Purpose
Step 10	(Optional) **show mpls ldp neighbor** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp neighbor`	Displays the status of the LDP session with its neighbors. This command can be run with various filters as well as with the brief option.

Setting Up LDP NSF Using Graceful Restart

Perform this task to set up NSF using LDP graceful restart.

LDP graceful restart is a way to enable NSF for LDP. The correct way to set up NSF using LDP graceful restart is to bring up LDP neighbors (link or targeted) with additional configuration related to graceful restart.

Before you begin

Stable router ID is required at either end of the link to ensure the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system will default to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **interface** *type interface-path-id*
4. **exit**
5. **graceful-restart**
6. **graceful-restart forwarding-state-holdtime** *seconds*
7. **graceful-restart reconnect-timeout** *seconds*
8. Use the **commit** or **end** command.
9. (Optional) **show mpls ldp parameters**
10. (Optional) **show mpls ldp neighbor**
11. (Optional) **show mpls ldp graceful-restart**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.

	Command or Action	Purpose

Step 3	**interface** *type interface-path-id*
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# interface POS 0/1/0/0`
	`RP/0/RP0/CPU0:router(config-ldp-if)#`
		Enters interface configuration mode for the LDP protocol.

Step 4	**exit** **Example:** `RP/0/RP0/CPU0:router(config-ldp-if)# exit`	Exits the current configuration mode.
Step 5	**graceful-restart** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# graceful-restart`	Enables the LDP graceful restart feature.
Step 6	**graceful-restart forwarding-state-holdtime** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# graceful-restart forwarding-state-holdtime 180`	Specifies the length of time that forwarding can keep LDP-installed forwarding states and rewrites, and specifies wh en the LDP control plane restarts. • After restart of the control plane, when the forwarding state holdtime expires, any previously installed LDP forwarding state or rewrite that is not yet refreshed is deleted from the forwarding. • Recovery time sent after restart is computed as the current remaining value of the forwarding state hold timer.
Step 7	**graceful-restart reconnect-timeout** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# graceful-restart reconnect-timeout 169`	Specifies the length of time a neighbor waits before restarting the node to reconnect before declaring an earlier graceful restart session as down. This command is used to start a timer on the peer (upon a neighbor restart). This timer is referred to as *Neighbor Liveness* timer.
Step 8	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

	Command or Action	Purpose
Step 9	(Optional) **show mpls ldp parameters** **Example:** `RP/0/RP0/CPU0:router # show mpls ldp parameters`	Displays all the current MPLS LDP parameters.
Step 10	(Optional) **show mpls ldp neighbor** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp neighbor`	Displays the status of the LDP session with its neighbors. This command can be run with various filters as well as with the brief option.
Step 11	(Optional) **show mpls ldp graceful-restart** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp graceful-restart`	Displays the status of the LDP graceful restart feature. The output of this command not only shows states of different graceful restart timers, but also a list of graceful restart neighbors, their state, and reconnect count.

Configuring Label Acceptance Control (Inbound Filtering)

Perform this task to configure LDP inbound label filtering.

Note By default, there is no inbound label filtering performed by LDP and thus an LSR accepts (and retains) all remote label bindings from all peers.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **label accept for** *prefix-acl* **from** *ip-address*
4. [**vrf** *vrf-name*] **address-family** { **ipv4**}
5. **label remote accept from** *ldp-id* **for** *prefix-acl*
6. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.

	Command or Action	Purpose
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters the MPLS LDP configuration mode.

Step 3	**label accept for** *prefix-acl* **from** *ip-address*
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# label accept for pfx_acl_1 from 192.168.1.1`
	`RP/0/RP0/CPU0:router(config-ldp)# label accept for pfx_acl_2 from 192.168.2.2`
		Configures inbound label acceptance for prefixes specified by prefix-acl from neighbor (as specified by its IP address).

Step 4	[**vrf** *vrf-name*] **address-family** { **ipv4**}
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# address-family ipv4`
	`RP/0/RP0/CPU0:router(config-ldp)# address-family ipv6`
		(Optional) Specifies a non-default VRF.
		Enables the LDP IPv4 or IPv6 address family.

Step 5	**label remote accept from** *ldp-id* **for** *prefix-acl* **Example:** `RP/0/RP0/CPU0:router(config-ldp-af)# label remote accept from 192.168.1.1:0 for pfx_acl_1`	Configures inbound label acceptance control for prefixes specified by prefix-acl from neighbor (as specified by its LDP ID).
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configuring LDP IGP Synchronization: ISIS

Perform this task to configure LDP IGP Synchronization under ISIS.

Note By default, there is no synchronization between LDP and ISIS.

SUMMARY STEPS

1. **configure**
2. **router isis** *instance-id*
3. **interface** *type interface-path-id*
4. **address-family** {**ipv4** } **unicast**
5. **mpls ldp sync**
6. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.

Step 2	**router isis** *instance-id*
	Example:
	`RP/0/RP0/CPU0:router(config)# router isis 100`
	`RP/0/RP0/CPU0:router(config-isis)#`
		Enables the Intermediate System-to-Intermediate System (IS-IS) routing protocol and defines an IS-IS instance.

Step 3	**interface** *type interface-path-id*
	Example:
	`RP/0/RP0/CPU0:router(config-isis)# interface POS 0/2/0/0`
	`RP/0/RP0/CPU0:router(config-isis-if)#`
		Configures the IS-IS protocol on an interface and enters ISIS interface configuration mode.

Step 4	**address-family** {**ipv4** } **unicast**
	Example:
	`RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast`
	`RP/0/RP0/CPU0:router(config-isis-if-af)#`
		Enters address family configuration mode for configuring IS-IS routing for a standard IP version 4 (IPv4) address prefix.

Step 5	**mpls ldp sync** **Example:** `RP/0/RP0/CPU0:router(config-isis-if-af)# mpls ldp sync`	Enables LDP IGP synchronization.

	Command or Action	Purpose
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configure Label Distribution Protocol Targeted Neighbor

LDP session between LSRs that are not directly connected is known as targeted LDP session. For LDP neighbors which are not directly connected, you must manually configure the LDP neighborship on both the routers.

Configuration Example

This example shows how to configure LDP for non-directly connected routers.

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# mpls ldp
RP/0/RSP0/CPU0:router(config-ldp)# router-id 192.0.2.1
RP/0/RSP0/CPU0:router(config-ldp)# neighbor 198.51.100.1:0 password encrypted 13061E010803
RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv4
RP/0/RSP0/CPU0:router(config-ldp-af)# discovery targeted-hello accept
RP/0/RSP0/CPU0:router(config-ldp-af)# neighbor 198.51.100.1 targeted
RP/0/RSP0/CPU0:router(config-ldp-af)# commit
```

Running Configuration

This section shows the LDP targeted neighbor running configuration.

```
mpls ldp
router-id 192.0.2.1
neighbor 198.51.100.1:0 password encrypted 13061E010803
address-family ipv4
  discovery targeted-hello accept
  neighbor 198.51.100.1 targeted
!
```

Verification

Verify LDP targeted neighbor configuration.

```
RP/0/RSP0/CPU0:router#show mpls ldp discovery
Wed Nov 28 04:30:31.862 UTC

Local LDP Identifier: 192.0.2.1:0
```

```
Discovery Sources:
  Targeted Hellos:  <<< targeted hellos based session
    192.0.2.1 -> 198.51.100.1(active/passive), xmit/recv   <<< both transmit and receive
of targeted hellos between the neighbors
      LDP Id: 198.51.100.1:0
          Hold time: 90 sec (local:90 sec, peer:90 sec)
          Established: Nov 28 04:19:55.340 (00:10:36 ago)

RP/0/RSP0/CPU0:router#show mpls ldp neigbhor
Wed Nov 28 04:30:38.272 UTC

Peer LDP Identifier: 198.51.100.1:0
  TCP connection: 198.51.100.1:0:13183 - 192.0.2.1:646; MD5 on
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 20/20; Downstream-Unsolicited
  Up time: 00:10:30
  LDP Discovery Sources:
    IPv4: (1)
      Targeted Hello (192.0.2.1 -> 198.51.100.1, active/passive) <<< targeted LDP based
session
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (4)
     198.51.100.1        10.0.0.1       172.16.0.1       192.168.0.1
    IPv6: (0)
```

Configuring Global Transport Address

Perform this task to configure global transport address for the IPv4 address family.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **address-family ipv4**
4. **discovery transport-address** *ip-address*
5. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**address-family ipv4** **Example:**	Enables LDP IPv4 address family.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-ldp)# address-family ipv4`	
Step 4	**discovery transport-address** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-ldp-af)# discovery transport-address 192.168.1.42`	Provides an alternative transport address for a TCP connection. • Default transport address advertised by an LSR (for TCP connections) to its peer is the router ID.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-af)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring IPv4 as Transport Preference

Perform this task to configure IPv4 as the preferred transport (overriding the default setting of IPv6 as preferred transport) to establish connection for a set of dual-stack peers.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **neighbor dual-stack transport-connection prefer ipv4 for-peers** *peer lsr-id*
4. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**neighbor dual-stack transport-connection prefer ipv4 for-peers** *peer lsr-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# neighbor dual-stack transport-connection prefer ipv4 for-peers 5.5.5.5`	Configures IPv4 as the preferred transport connection for the specified peer.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring LDP Discovery for Active Targeted Hellos

Perform this task to configure LDP discovery for active targeted hellos.

Note The active side for targeted hellos initiates the unicast hello toward a specific destination.

Before you begin

These prerequisites are required to configure LDP discovery for active targeted hellos:

- Stable router ID is required at either end of the targeted session. If you do not assign a router ID to the routers, the system will default to the global router ID. Please note that default router IDs are subject to change and may cause an unstable discovery.
- One or more MPLS Traffic Engineering tunnels are established between non-directly connected LSRs.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. [**vrf** *vrf-name*] **router-id** *ip-address lsr-id*
4. **interface** *type interface-path-id*
5. Use the **commit** or **end** command.
6. (Optional) **show mpls ldp discovery**
7. (Optional) **show mpls ldp vrf** *vrf-name* **discovery**
8. (Optional) **show mpls ldp vrf all discovery summary**
9. (Optional) **show mpls ldp vrf all discovery brief**
10. (Optional) **show mpls ldp vrf all ipv4 discovery summary**
11. (Optional) **show mpls ldp discovery summary all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	[**vrf** *vrf-name*] **router-id** *ip-address lsr-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# router-id 192.168.70.1`	(Optional) Specifies a non-default VRF. Specifies the router ID of the local node. In Cisco IOS XR software, the router ID is specified as an interface name or IP address or LSR ID. By default, LDP uses the global router ID (configured by global router ID process).

	Command or Action	Purpose
Step 4	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface tunnel-te 12001`	Enters interface configuration mode for the LDP protocol.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 6	(Optional) **show mpls ldp discovery** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp discovery`	Displays the status of the LDP discovery process. This command, without an interface filter, generates a list of interfaces over which the LDP discovery process is running. The output information contains the state of the link (xmt/rcv hellos), local LDP identifier, the discovered peer's LDP identifier, and holdtime values.
Step 7	(Optional) **show mpls ldp vrf** *vrf-name* **discovery** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf red discovery`	Displays the status of the LDP discovery process for the specified VRF.
Step 8	(Optional) **show mpls ldp vrf all discovery summary** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs.
Step 9	(Optional) **show mpls ldp vrf all discovery brief** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery brief`	Displays the brief status of the LDP discovery process for all VRFs.
Step 10	(Optional) **show mpls ldp vrf all ipv4 discovery summary** **Example:**	Displays the summarized status of the LDP discovery process for all VRFs for the IPv4 address family.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# show mpls ldp vrf all ipv4 discovery summary`	
Step 11	(Optional) **show mpls ldp discovery summary all** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp discovery summary all`	Displays the aggregate summary across all the LDP discovery processes.

Configuring LDP Discovery for Passive Targeted Hellos

Perform this task to configure LDP discovery for passive targeted hellos.

A passive side for targeted hello is the destination router (tunnel tail), which passively waits for an incoming hello message. Because targeted hellos are unicast, the passive side waits for an incoming hello message to respond with hello toward its discovered neighbor.

Before you begin

Stable router ID is required at either end of the link to ensure that the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. [**vrf** *vrf-name*] **router-id** *ip-address lsr-id*
4. **discovery targeted-hello accept**
5. Use the **commit** or **end** command.
6. (Optional) **show mpls ldp discovery**
7. (Optional) **show mpls ldp vrf** *vrf-name* **discovery**
8. (Optional) **show mpls ldp vrf all discovery summary**
9. (Optional) **show mpls ldp vrf all discovery brief**
10. (Optional) **show mpls ldp vrf all ipv4 discovery summary**
11. (Optional) **show mpls ldp discovery summary all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.

	Command or Action	Purpose
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	[**vrf** *vrf-name*] **router-id** *ip-address lsr-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# router-id 192.168.70.1`	(Optional) Specifies a non-default VRF. Specifies the router ID of the local node. • In Cisco IOS XR software, the router ID is specified as an interface IP address or LSR ID. By default, LDP uses the global router ID (configured by global router ID process).
Step 4	**discovery targeted-hello accept** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# discovery targeted-hello accept`	Directs the system to accept targeted hello messages from any source and activates passive mode on the LSR for targeted hello acceptance. • This command is executed on the receiver node (with respect to a given MPLS TE tunnel). • You can control the targeted-hello acceptance using the **discovery targeted-hello accept** command.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 6	(Optional) **show mpls ldp discovery** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp discovery`	Displays the status of the LDP discovery process. This command, without an interface filter, generates a list of interfaces over which the LDP discovery process is running. The output information contains the state of the link (xmt/rcv hellos), local LDP identifier, the discovered peer's LDP identifier, and holdtime values.
Step 7	(Optional) **show mpls ldp vrf** *vrf-name* **discovery** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf red discovery`	Displays the status of the LDP discovery process for the specified VRF.

	Command or Action	Purpose
Step 8	(Optional) **show mpls ldp vrf all discovery summary** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs.
Step 9	(Optional) **show mpls ldp vrf all discovery brief** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery brief`	Displays the brief status of the LDP discovery process for all VRFs.
Step 10	(Optional) **show mpls ldp vrf all ipv4 discovery summary** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all ipv4 discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs for the IPv4 address family.
Step 11	(Optional) **show mpls ldp discovery summary all** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp discovery summary all`	Displays the aggregate summary across all the LDP discovery processes.

Configuring LDP Discovery Over a Link

Perform this task to configure LDP discovery over a link.

Note There is no need to enable LDP globally.

Before you begin

A stable router ID is required at either end of the link to ensure the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system will default to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. [**vrf** *vrf-name*] **router-id** *ip-address lsr-id*
4. **interface** *type interface-path-id*
5. Use the **commit** or **end** command.
6. (Optional) **show mpls ldp discovery**

7. (Optional) **show mpls ldp vrf** *vrf-name* **discovery**
8. (Optional) **show mpls ldp vrf all discovery summary**
9. (Optional) **show mpls ldp vrf all discovery brief**
10. (Optional) **show mpls ldp vrf all ipv4 discovery summary**
11. (Optional) **show mpls ldp discovery summary all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	[**vrf** *vrf-name*] **router-id** *ip-address lsr-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# router-id 192.168.70.1`	(Optional) Specifies a non-default VRF. Specifies the router ID of the local node. • In Cisco IOS XR software, the router ID is specified as an interface name or IP address. By default, LDP uses the global router ID (configured by the global router ID process).

Step 4	**interface** *type interface-path-id*
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# interface tunnel-te 12001`
	`RP/0/RP0/CPU0:router(config-ldp-if)#`
		Enters interface configuration mode for the LDP protocol. Interface type must be Tunnel-TE.

Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 6	(Optional) **show mpls ldp discovery** **Example:**	Displays the status of the LDP discovery process. This command, without an interface filter, generates a list of interfaces over which the LDP discovery process is

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# show mpls ldp discovery`	running. The output information contains the state of the link (xmt/rcv hellos), local LDP identifier, the discovered peer's LDP identifier, and holdtime values.
Step 7	(Optional) **show mpls ldp vrf** *vrf-name* **discovery** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf red discovery`	Displays the status of the LDP discovery process for the specified VRF.
Step 8	(Optional) **show mpls ldp vrf all discovery summary** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs.
Step 9	(Optional) **show mpls ldp vrf all discovery brief** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery brief`	Displays the brief status of the LDP discovery process for all VRFs.
Step 10	(Optional) **show mpls ldp vrf all ipv4 discovery summary** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all ipv4 discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs for the IPv4 address family.
Step 11	(Optional) **show mpls ldp discovery summary all** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp discovery summary all`	Displays the aggregate summary across all the LDP discovery processes.

Configuring Downstream on Demand

By default, LDP uses downstream unsolicited mode in which label advertisements for all routes are received from all LDP peers. The downstream on demand feature adds support for downstream-on-demand mode, where the label is not advertised to a peer, unless the peer explicitly requests it. At the same time, since the peer does not automatically advertise labels, the label request is sent whenever the next-hop points out to a peer that no remote label has been assigned.

In downstream on demand configuration, an ACL is used to specify the set of peers for downstream on demand mode. For down stream on demand to be enabled, it needs to be configured on both peers of the session. If only one peer in the session has downstream-on-demand feature configured, then the session does not use downstream-on-demand mode.

Configuration Example

This example shows how to configure LDP Downstream on Demand.

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# session downstream-on-demand with ACL1
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

Configuring LDP Link: Example

The example shows how to configure LDP link parameters.

```
mpls ldp
 interface pos 0/1/0/0
 !
!

show mpls ldp discovery
```

Configuring Label Distribution Protocol Nonstop Routing

LDP nonstop routing (NSR) functionality makes failures, such as Route Processor (RP) or Distributed Route Processor (DRP) fail over, invisible to routing peers with minimal to no disruption of convergence performance. By default, NSR is globally enabled on all LDP sessions except AToM.

A disruption in service may include any of these events:

- Route processor (RP) or distributed route processor (DRP) failover
- LDP process restart
- Minimum disruption restart (MDR)

Note Unlike graceful restart functionality, LDP NSR does not require protocol extensions and does not force software upgrades on other routers in the network, nor does LDP NSR require peer routers to support NSR. L2VPN configuration is not supported on NSR. Process failures of active LDP results in session loss and, as a result, NSR cannot be provided unless RP switchover is configured as a recovery action.

Configuration Example

This example shows how to configure LDP Non-Stop Routing.

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# nsr
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

Verification

```
RP/0/RP0/CPU0:Router# show mpls ldp nsr summary
Mon Dec  7 04:02:16.259 UTC
```

```
Sessions:
Total: 1, NSR-eligible: 1, Sync-ed: 0
(1 Ready)
```

Configure Session Protection

Configuration Example

As per the configuration, LDP session protection for peers specified by peer-acl is configured to maximum duration of 60 seconds.

```
Router(config)#mpls ldp
Router(config-ldp)#session protection for peer_acl_1 duration 60
Router(config-ldp)#commit
```

Configuring Local Label Allocation Control

Perform this task to configure label allocation control.

Note By default, local label allocation control is disabled and all non-BGP prefixes are assigned local labels.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **label allocate for** *prefix-acl*
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters the MPLS LDP configuration mode.
Step 3	**label allocate for** *prefix-acl* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# label allocate for pfx_acl_1`	Configures label allocation control for prefixes as specified by prefix-acl.

	Command or Action	Purpose
Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configuring Transport Preference Maximum Wait Time

Perform this task to configure the maximum time (in seconds) the preferred address family connection must wait to establish transport connection before resorting to non-preferred address family.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **neighbor dual-stack transport-connection max-wait** *seconds*
4. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**neighbor dual-stack transport-connection max-wait** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# neighbor dual-stack transport-connection max-wait 5`	Configures the maximum wait time.

	Command or Action	Purpose

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring MPLS Label Security

Perform this task to configure the MPLS label security on an interface

Configuration Example

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# mpls label-security rpf
Router(config-if)# commit
```

Verification

Use the **show mpls forwarding label-security interface** command to view MPLS label security configuration on an interface.

Disabling Implicit IPv4

Perform this task to disable the implicitly enabled IPv4 address family for default VRF.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **default-vrf implicit-ipv4 disable**
4. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**default-vrf implicit-ipv4 disable** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# default-vrf implicit-ipv4 disable`	Disables the implicitly enabled IPv4 address family for default VRF.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Disabling LDP Auto-Configuration

Perform this task to disable IGP auto-configuration.

You can disable auto-configuration on a per-interface basis. This lets LDP enable all IGP interfaces except those that are explicitly disabled.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **interface** *type interface-path-id*
4. **igp auto-config disable**
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.

Step 2	**mpls ldp**
	Example:
	`RP/0/RP0/CPU0:router(config)# mpls ldp`
	`RP/0/RP0/CPU0:router(config-ldp)#`
		Enters the MPLS LDP configuration mode.

Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface pos 0/6/0/0`	Enters interface configuration mode and configures an interface.
Step 4	**igp auto-config disable** **Example:** `RP/0/RP0/CPU0:router(config-ldp-if)# igp auto-config disable`	Disables auto-configuration on the specified interface.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Disabling LDP IGP Synchronization: OSPF

Perform this task to disable LDP IGP Synchronization under OSPF.

You can disable LDP IGP synchronization on a per-interface basis. This lets LDP enable all IGP interfaces except those that are explicitly disabled.

SUMMARY STEPS

1. **configure**
2. **router ospf** *process-name*
3. Use one of the following commands:
   - **area** *area-id* **mpls ldp sync disable**
   - **area** *area-id* **interface** *name* **mpls ldp sync disable**
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**router ospf** *process-name* **Example:** `RP/0/RP0/CPU0:router(config)# router ospf 109`	Identifies the OSPF routing process and enters OSPF configuration mode.

Step 3	Use one of the following commands:
	• **area** *area-id* **mpls ldp sync disable**
	• **area** *area-id* **interface** *name* **mpls ldp sync disable**
	Example:
	`RP/0/RP0/CPU0:router(config-ospf)# area 1 mpls ldp sync disable`
	`RP/0/RP0/CPU0:router(config-ospf)# area 1 interface POS 0/2/0/0 mpls ldp sync disable`
		Disables LDP IGP synchronization on an interface.

Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session.

	Command or Action	Purpose
		• **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Disabling MLDP

Perform this task to disable MLDP on Label Distribution Protocol (LDP) enabled interfaces.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **interface** *type interface-path-id*
4. **address-family** {**ipv4** }
5. **igp mldp disable**
6. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface POS 0/1/0/0`	Enters interface configuration mode for the LDP protocol.
Step 4	**address-family** {**ipv4** } **Example:** `RP/0/RP0/CPU0:router(config-ldp-if)# address-family ipv4` or	Enables the LDP IPv4 address family.

	Command or Action	Purpose
Step 5	**igp mldp disable** **Example:** `RP/0/RP0/CPU0:router(config-ldp-if-af)# igp mldp disable`	Disables MLDP.
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Enabling LDP Auto-Configuration for a Specified OSPF Instance

Perform this task to enable IGP auto-configuration globally for a specified OSPF process name.

You can disable auto-configuration on a per-interface basis. This lets LDP enable all IGP interfaces except those that are explicitly disabled.

Note This feature is supported for IPv4 unicast family in default VRF only.

SUMMARY STEPS

1. **configure**
2. **router ospf** *process-name*
3. **mpls ldp auto-config**
4. **area** *area-id*
5. **interface** *type interface-path-id*
6. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.

	Command or Action	Purpose

Step 2	**router ospf** *process-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# router ospf 190`
	`RP/0/RP0/CPU0:router(config-ospf)#`
		Enters a uniquely identifiable OSPF routing process. The process name is any alphanumeric string no longer than 40 characters without spaces.

Step 3	**mpls ldp auto-config** **Example:** `RP/0/RP0/CPU0:router(config-ospf)# mpls ldp auto-config`	Enables LDP auto-configuration.
Step 4	**area** *area-id* **Example:** `RP/0/RP0/CPU0:router(config-ospf)# area 8`	Configures an OSPF area and identifier. ***area-id*** Either a decimal value or an IP address.
Step 5	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar)# interface pos 0/6/0/0`	Enables LDP auto-configuration on the specified interface. **Note** LDP configurable limit for maximum number of interfaces does not apply to IGP auto-configuration interfaces.
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Related Topics

Disabling LDP Auto-Configuration, on page 56

Enabling LDP Auto-Configuration in an Area for a Specified OSPF Instance

Perform this task to enable IGP auto-configuration in a defined area with a specified OSPF process name.

You can disable auto-configuration on a per-interface basis. This lets LDP enable all IGP interfaces except those that are explicitly disabled.

Note This feature is supported for IPv4 unicast family in default VRF only.

SUMMARY STEPS

1. **configure**
2. **router ospf** *process-name*
3. **area** *area-id*
4. **mpls ldp auto-config**
5. **interface** *type interface-path-id*
6. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.

Step 2	**router ospf** *process-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# router ospf 100`
	`RP/0/RP0/CPU0:router(config-ospf)#`
		Enters a uniquely identifiable OSPF routing process. The process name is any alphanumeric string no longer than 40 characters without spaces.

Step 3	**area** *area-id*
	Example:
	`RP/0/RP0/CPU0:router(config-ospf)# area 8`
	`RP/0/RP0/CPU0:router(config-ospf-ar)#`
		Configures an OSPF area and identifier.
		area-id
		Either a decimal value or an IP address.

Step 4	**mpls ldp auto-config** **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar)# mpls ldp auto-config`	Enables LDP auto-configuration.

Step 5	**interface** *type interface-path-id*
	Example:
	`RP/0/RP0/CPU0:router(config-ospf-ar)# interface pos 0/6/0/0`
	`RP/0/RP0/CPU0:router(config-ospf-ar-if)`
		Enables LDP auto-configuration on the specified interface. The LDP configurable limit for maximum number of interfaces does not apply to IGP auto-config interfaces.

Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session.

	Command or Action	Purpose
		end —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Related Topics

Disabling LDP Auto-Configuration, on page 56

Implicit IPv4 Disable

The LDP configuration model was changed with the introduction of explicit address family enabling under LDP (VRF) global and LDP (VRF) interfaces. However, in order to support backward compatibility, the old configuration model was still supported for default VRF. There was, however, no option to disable the implicitly enabled IPv4 address family under default VRF's global or interface level.

A new configuration **mpls ldp default-vrf implicit-ipv4 disable** is now available to the user to disable the implicitly enabled IPv4 address family for the default VRF. The new configuration provides a step towards migration to new configuration model for the default VRF that mandates enabling address family explicitly. This means that if the new option is configured, the user has to explicitly enable IPv4 address family for default VRF global and interface levels. It is recommended to migrate to this explicitly enabled IPv4 configuration model.

For detailed configuration steps, see Disabling Implicit IPv4, on page 55

Running Configuration

This section shows the LDP targeted neighbor running configuration.

```
mpls ldp
router-id 192.0.2.1
neighbor 198.51.100.1:0 password encrypted 13061E010803
address-family ipv4
  discovery targeted-hello accept
  neighbor 198.51.100.1 targeted
!
```

Verification

Verify LDP targeted neighbor configuration.

```
RP/0/RSP0/CPU0:router#show mpls ldp discovery
Wed Nov 28 04:30:31.862 UTC

Local LDP Identifier: 192.0.2.1:0
Discovery Sources:
  Targeted Hellos:  <<< targeted hellos based session
    192.0.2.1 -> 198.51.100.1(active/passive), xmit/recv   <<< both transmit and receive
of targeted hellos between the neighbors
      LDP Id: 198.51.100.1:0
```

```
          Hold time: 90 sec (local:90 sec, peer:90 sec)
          Established: Nov 28 04:19:55.340 (00:10:36 ago)

RP/0/RSP0/CPU0:router#show mpls ldp neigbhor
Wed Nov 28 04:30:38.272 UTC

Peer LDP Identifier: 198.51.100.1:0
  TCP connection: 198.51.100.1:0:13183 - 192.0.2.1:646; MD5 on
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 20/20; Downstream-Unsolicited
  Up time: 00:10:30
  LDP Discovery Sources:
    IPv4: (1)
      Targeted Hello (192.0.2.1 -> 198.51.100.1, active/passive) <<< targeted LDP based
session
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (4)
     198.51.100.1        10.0.0.1       172.16.0.1       192.168.0.1
    IPv6: (0)
```

Verify IP LDP Fast Reroute Loop Free Alternate: Example

Configuration Examples for Implementing MPLS LDP

These configuration examples are provided to implement LDP:

Configuring LDP Discovery for Targeted Hellos: Example

The examples show how to configure LDP Discovery to accept targeted hello messages.

Active (tunnel head)

```
mpls ldp
   router-id 192.168.70.1
   interface tunnel-te 12001
   !
  !
```

Passive (tunnel tail)

```
mpls ldp
   router-id 192.168.70.2
   discovery targeted-hello accept
  !
```

Configure IP LDP Fast Reroute Loop Free Alternate: Examples

This example shows how to configure LFA FRR with default tie-break configuration:

```
router isis TEST
 net 49.0001.0000.0000.0001.00
 address-family ipv4 unicast
```

```
  metric-style wide

 interface GigabitEthernet0/6/0/13
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
   # primary path GigabitEthernet0/6/0/13 will exclude the interface
   # GigabitEthernet0/6/0/33 in LFA backup path computation.
   fast-reroute per-prefix exclude interface GigabitEthernet0/6/0/33
!
 interface GigabitEthernet0/6/0/23
  point-to-point
  address-family ipv4 unicast
!
 interface GigabitEthernet0/6/0/24
  point-to-point
  address-family ipv4 unicast
!
 interface GigabitEthernet0/6/0/33
  point-to-point
  address-family ipv4 unicast
!
```

This example shows how to configure TE tunnel as LFA backup:

```
router isis TEST
 net 49.0001.0000.0000.0001.00
 address-family ipv4 unicast
  metric-style wide

 interface GigabitEthernet0/6/0/13
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
   # primary path GigabitEthernet0/6/0/13 will exclude the interface
   # GigabitEthernet0/6/0/33 in LFA backup path computation. TE tunnel 1001
   # is using the link GigabitEthernet0/6/0/33.
   fast-reroute per-prefix exclude interface GigabitEthernet0/6/0/33
   fast-reroute per-prefix lfa-candidate interface tunnel-te1001
!
 interface GigabitEthernet0/6/0/33
  point-to-point
  address-family ipv4 unicast
!
```

This example shows how to configure LFA FRR with configurable tie-break configuration:

```
router isis TEST
 net 49.0001.0000.0000.0001.00
 address-family ipv4 unicast
  metric-style wide
  fast-reroute per-prefix tiebreaker ?
  downstream            Prefer backup path via downstream node
  lc-disjoint           Prefer line card disjoint backup path
  lowest-backup-metric  Prefer backup path with lowest total metric
  node-protecting       Prefer node protecting backup path
  primary-path          Prefer backup path from ECMP set
  secondary-path        Prefer non-ECMP backup path

  fast-reroute per-prefix tiebreaker lc-disjoint index ?
```

```
  <1-255>  Index
  fast-reroute per-prefix tiebreaker lc-disjoint index 10

Sample configuration:

router isis TEST
 net 49.0001.0000.0000.0001.00
 address-family ipv4 unicast
  metric-style wide
  fast-reroute per-prefix tiebreaker downstream index 60
  fast-reroute per-prefix tiebreaker lc-disjoint index 10
  fast-reroute per-prefix tiebreaker lowest-backup-metric index 40
  fast-reroute per-prefix tiebreaker node-protecting index 30
  fast-reroute per-prefix tiebreaker primary-path index 20
  fast-reroute per-prefix tiebreaker secondary-path index 50
!
interface GigabitEthernet0/6/0/13
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
!
 interface GigabitEthernet0/1/0/13
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
!
 interface GigabitEthernet0/3/0/0.1
  point-to-point
  address-family ipv4 unicast
!
 interface GigabitEthernet0/3/0/0.2
  point-to-point
  address-family ipv4 unicast
```

Configuring Local Label Allocation Control: Example

The example shows how to configure local label allocation control.

```
  mpls ldp
   label
  allocate for pfx_acl_1
   !
  !
```

Configuring LDP with Graceful Restart: Example

The example shows how to enable LDP with graceful restart on the POS interface 0/2/0/0.

```
  mpls ldp
   graceful-restart
   interface pos0/2/0/0
  !
```

Configuring LDP Forwarding: Example

The example shows how to configure LDP forwarding.

```
mpls ldp
  address-family ipv4
  label local advertise explicit-null
 !

 show mpls ldp forwarding
 show mpls forwarding
```

Configuring LDP Nonstop Forwarding with Graceful Restart: Example

The example shows how to configure LDP nonstop forwarding with graceful restart.

```
 mpls ldp
 log
 graceful-restart
 !
  graceful-restart
  graceful-restart forwarding state-holdtime 180
  graceful-restart reconnect-timeout 15
  interface pos0/1/0/0
 !

 show mpls ldp graceful-restart
 show mpls ldp neighbor gr
 show mpls ldp forwarding
 show mpls forwarding
```

Configuring Label Acceptance (Inbound Filtering): Example

The example shows how to configure inbound label filtering.

```
mpls ldp
  label
 accept
  for pfx_acl_2 from 192.168.2.2
 !
  !
 !
```

```
 mpls ldp
  address-family ipv4
   label remote accept from 192.168.1.1:0 for pfx_acl_2
   !
  !
 !
```

Configuring LDP Discovery: Example

The example shows how to configure LDP discovery parameters.

```
mpls ldp
 router-id 192.168.70.1
 discovery hello holdtime 15
 discovery hello interval 5
!

show mpls ldp parameters
show mpls ldp discovery
```

Configuring LDP Auto-Configuration: Example

The example shows how to configure the IGP auto-configuration feature globally for a specific OSPF interface ID.

```
router ospf 100
   mpls ldp auto-config
   area 0
    interface pos 1/1/1/1
```

The example shows how to configure the IGP auto-configuration feature on a given area for a given OSPF interface ID.

```
router ospf 100
 area 0
  mpls ldp auto-config
  interface pos 1/1/1/1
```

Configuring LDP Neighbors: Example

The example shows how to disable label advertisement.

```
mpls ldp
    router-id 192.168.70.1
    neighbor 10.0.0.1 password encrypted 110A1016141E
    neighbor 172.16.0.1 implicit-withdraw
!
```

Configuring LDP IGP Synchronization—ISIS: Example

The example shows how to configure LDP IGP synchronization.

```
router isis 100
 interface POS 0/2/0/0
address-family ipv4 unicast
mpls ldp sync
!
 !
!
mpls ldp
 igp sync delay 30
!
```

Configuring LDP IGP Synchronization—OSPF: Example

The example shows how to configure LDP IGP synchronization for OSPF.

```
router ospf 100
mpls ldp sync
!
mpls ldp
 igp sync delay 30
!
```

Label Distribution Protocol Interior Gateway Protocol Auto-configuration

Interior Gateway Protocol (IGP) auto-configuration allows you to automatically configure LDP on all interfaces associated with a specified IGP interface; for example, when LDP is used for transport in the core network. However, there needs to be one IGP set up to enable LDP auto-configuration.

Typically, LDP assigns and advertises labels for IGP routes and must often be enabled on all active interfaces by an IGP. Without IGP auto-configuration, you must define the set of interfaces under LDP, a procedure that is time-intensive and error-prone.

Controlling State Advertisements in an mLDP-Only Setup

Table 3: Feature History Table

Feature Name	Release Information	Description
Controlling State Advertisements in an mLDP-Only Setup	Release 7.5.2	In conformance with RFC 7473, you can control state advertisements of non-negotiated Label Distribution Protocol (LDP) applications in a Multipoint LDP (mLDP)-only environment. In such an environment, participating routers don't need to exchange any unicast binding information. As a result, the flow of LDP state information in an mLDP-only setup is faster. Also, when routers come up after a network event, the network convergence time is shorter.

This function explains controlling of state advertisements of non-negotiated Label Distribution Protocol (LDP) applications. This implementation is in conformance with RFC 7473 (Controlling State Advertisements of Non-negotiated LDP Applications).

The main purpose of documenting this function is to use it in a Multipoint LDP (mLDP)-only environment, wherein participating routers don’t need to exchange any unicast binding information.

Non-Negotiated LDP Applications

The LDP capabilities framework enables LDP applications' capabilities exchange and negotiation, thereby enabling LSRs to send necessary LDP state. However, for the applications that existed prior to the definition of the framework (called *non-negotiated* LDP applications), there is no capability negotiation done. When an LDP session comes up, an LDP speaker may unnecessarily advertise its local state (without waiting for any capabilities exchange and negotiation). In other words, even when the peer session is established for Multipoint LDP (mLDP), the LSR advertises the state for these early LDP applications.

One example is *IPv4/IPv6 Prefix LSPs Setup* (used to set up Label Switched Paths [LSPs] for IP prefixes). Another example is *L2VPN P2P FEC 128 and FEC 129 PWs Signaling* (an LDP application that signals point-to-point [P2P] Pseudowires [PWs] for Layer 2 Virtual Private Networks [L2VPNs]).

In an mLDP-only setup, you can disable these non-negotiated LDP applications and avoid unnecessary LDP state advertisement. An LDP speaker that only runs mLDP announces to its peer(s) its disinterest (or non-support) in non-negotiated LDP applications. That is, it announces to its peers its disinterest to set up IP Prefix LSPs or to signal L2VPN P2P PW, at the time of session establishment.

Upon receipt of such a capability, the receiving LDP speaker, if supporting the capability, disables the advertisement of the state related to the application towards the sender of the capability. This new capability can also be sent later in a Capability message, either to disable a previously enabled application's state advertisement, or to enable a previously disabled application's state advertisement.

As a result, the flow of LDP state information in an mLDP-only setup is faster. When routers come up after a network event, the network convergence time is fast too.

IP Address Bindings In An mLDP Setup

An LSR typically uses peer IP address(es) to map an IP routing next hop to an LDP peer in order to implement its control plane procedures. mLDP uses a peer's IP address(es) to determine its upstream LSR to reach the root node, and to select the forwarding interface towards its downstream LSR. Hence, in an mLDP-only network, while it is desirable to disable advertisement of label bindings for IP (unicast) prefixes, disabling advertisement of IP address bindings will break mLDP functionality.

Uninteresting State - For the *Prefix-LSP* LDP application, *uninteresting* state refers to any state related to IP Prefix FEC, such as FEC label bindings and LDP Status. IP address bindings are not considered as an *uninteresting* state.

For the P2P-PW application LDP application, *uninteresting* state refers to any state related to P2P PW FEC 128 or FEC 129, such as FEC label bindings, MAC address withdrawal, and LDP PW status.

Control State Advertisement

To control advertisement of *uninteresting* state of non-negotiated LDP applications, the capability parameter TLV *State Advertisement Control Capability* is used. This TLV is only present in the Initialization and Capability messages, and the TLV can hold one or more State Advertisement Control (SAC) Elements.

As an example, consider two LSRs, S (LDP speaker) and P (LDP peer), that support all non-negotiated applications. S is participating (or set to participate) in an mLDP-only setup. Pointers for this scenario:

- By default, the LSRs will advertise state for all LDP applications to their peers, as soon as an LDP session is established.
- The **capabilities sac mldp-only** function is enabled on S.
- P receives an update from S via a Capability message that specifies to disable all four non-negotiated applications states.

- P's outbound policy towards S blocks and disables state for the unneeded applications.
- S only receives mLDP advertisements from specific mLDP-participating peers.

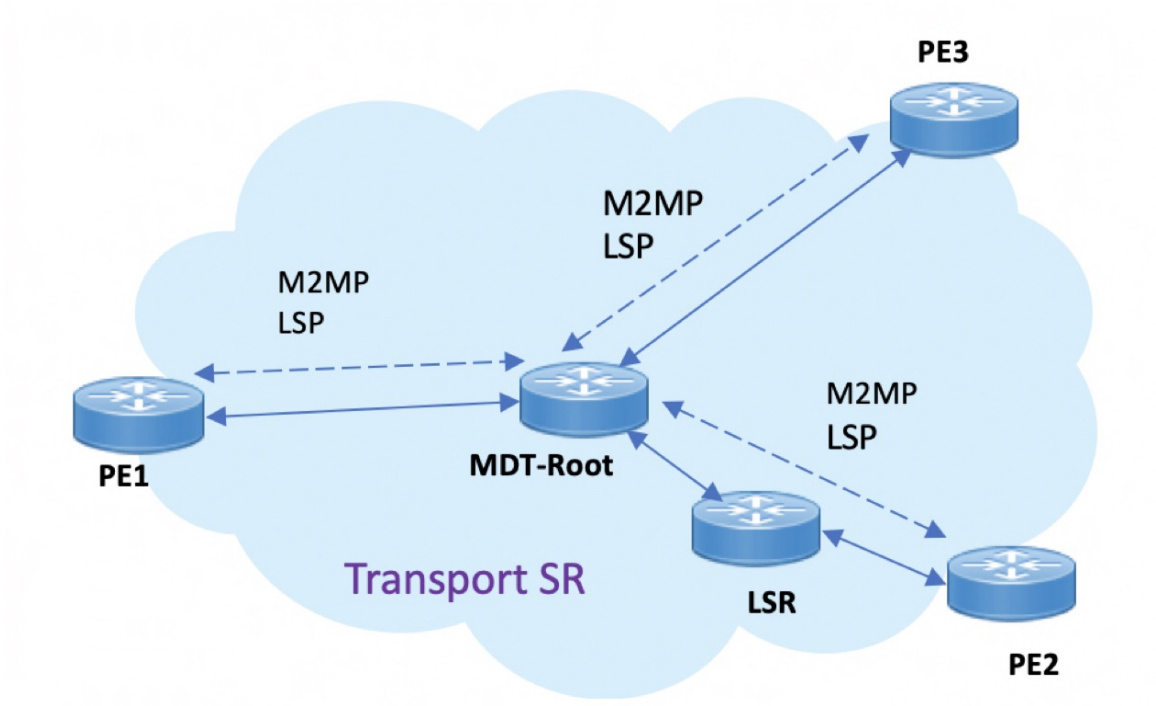
Use Cases For Controlling State Advertisements

Two use cases are explained, **mLDP-Based MVPN** and **Disable Prefix-LSPs On An L2VPN/PW tLDP Session**.

mLDP-Based MVPN

A sample topology and relevant configurations are noted below.

Figure 7: mLDP-Based MVPN Over Segment Routing



- The topology represents an MVPN profile 1 where an mLDP-based MVPN service is deployed over a Segment Routing core setup
- mLDP is required to signal MP2MP LSPs, whereas SR handles the transport.
- SAC capabilities are used to signal *mLDP-only* capability, which blocks unrequired unicast IPv4, IPv6, FEC128, and FEC129 related label binding advertisements.
- The **mldp-only** option is enabled on PE routers and P routers to remove unwanted advertisements.

Configuration

PE1 Configuration

Configure mLDP SAC capability on PE1.

```
PE1(config)# mpls ldp
PE1(config-ldp)# capabilities sac mldp-only
PE1(config-ldp)# commit
```

PE2 Configuration

Configure mLDP SAC capability on PE2.

```
PE2(config)# mpls ldp
PE2(config-ldp)# capabilities sac mldp-only
PE2(config-ldp)# commit
```

Verification

LDP peers (PE1 and PE2) are configured with **mldp-only** option, disabling all other SAC capabilities.

```
PE1# show running-config mpls ldp

mpls ldp
   capabilities sac mldp-only
   mldp
        address-family ipv4
        !

PE2# show running-config mpls ldp

mpls ldp
   capabilities sac mldp-only
   mldp
        address-family ipv4
        !
```

On PE1, verify PE2's SAC capabilities:

```
PE1# show mpls ldp neighbor 209.165.201.20 capabilities detail

Peer LDP Identifier: 209.165.201.20:0
  Capabilities:
    Sent:
      0x508  (MP: Point-to-Multipoint (P2MP))
      0x509  (MP: Multipoint-to-Multipoint (MP2MP))
      0x50b  (Typed Wildcard FEC)
      0x50d  (State Advertisement Control)
        [ {IPv4-disable}{IPv6-disable}{FEC128-disable}{FEC129-disable} ](length 4)
    Received:
      0x508  (MP: Point-to-Multipoint (P2MP))
      0x509  (MP: Multipoint-to-Multipoint (MP2MP))
      0x50b  (Typed Wildcard FEC)
      0x50d  (State Advertisement Control)
        [ {IPv4-disable}{IPv6-disable}{FEC128-disable}{FEC129-disable} ](length 4)
```

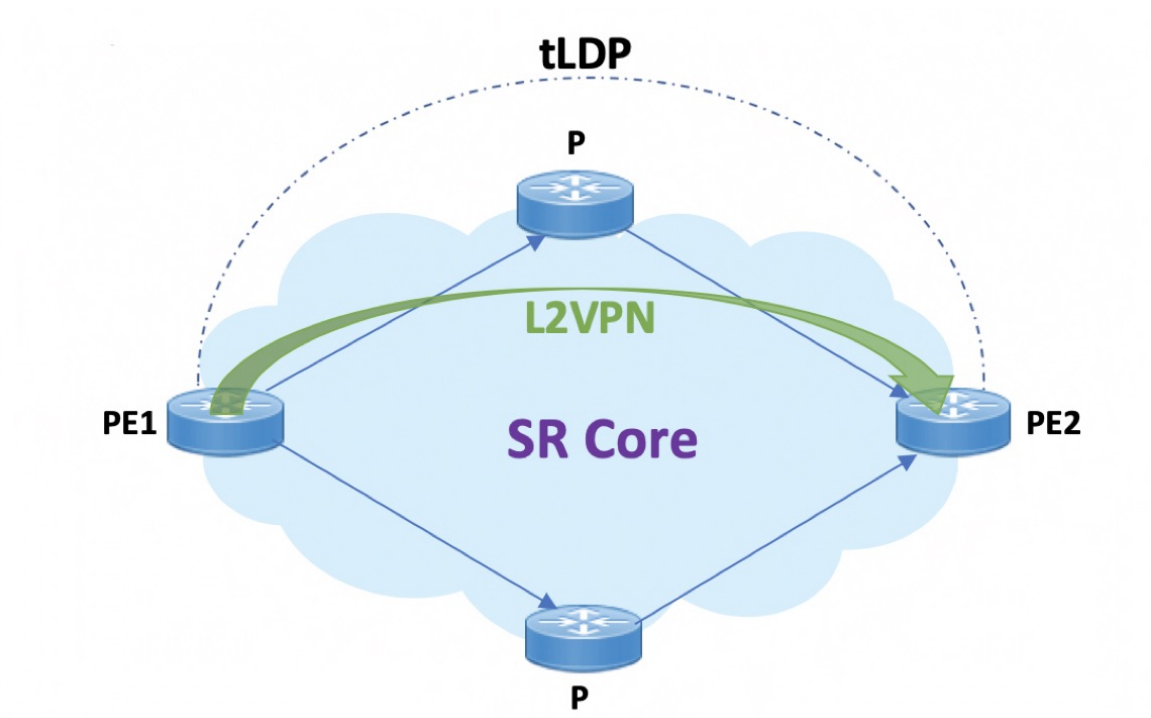
Capabilities Sent shows that **mldp-only** option disables all other advertisements.

Capabilities Received shows that **mldp-only** is enabled on peer PE2 too.

Disable Prefix-LSPs On An L2VPN/PW tLDP Session

A sample topology and relevant configurations are noted below.

Figure 8: L2VPN Xconnect Service Over Segment Routing



- The topology represents an L2VPN Xconnect service over a Segment Routing core setup.
- By default, Xconnect uses tLDP to signal service labels to remote PEs.
- By default, tLDP not only signals the service label, but also known (IPv4 and IPv6) label bindings to the tLDP peer, which is not required.
- The LDP SAC capabilities is an optional configuration enabled under LDP, and users can block IPv4 and IPv6 label bindings by applying configurations on PE1 and PE2.

Configuration

PE1 Configuration

Disable IPv4 prefix LSP binding advertisements on PE1:

```
PE1(config)# mpls ldp capabilities sac ipv4-disable
PE1(config)# commit
```

Disable IPv6-prefix LSP binding advertisements on PE1:

```
PE1(config)# mpls ldp capabilities sac ipv4-disable ipv6-disable
PE1(config)# commit
```

Note Whenever you disable a non-negotiated LDP application state on a router, you must include previously disabled non-negotiated LDP applications too, in the same command line. If not, the latest configuration overwrites the existing ones. You can see that ipv4-disable is added again, though it was already disabled.

PE2 Configuration

Enable SAC capability awareness on PE2, and make PE2 stop sending IPv4 prefix LSP binding advertisements to PE1:

```
PE2(config)#mpls ldp capabilities sac
PE2(config)#commit
```

Verification

On PE1, verify PE2's SAC capabilities:

```
PE1# show mpls ldp neighbor 198.51.100.1 detail

Peer LDP Identifier: 198.51.100.1:0
  TCP connection: 198.51.100.1:29132 - 192.0.2.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 14/14; Downstream-Unsolicited
  Up time: 00:03:30
  LDP Discovery Sources:
    IPv4: (1)
      Targeted Hello (192.0.2.1 -> 198.51.100.1, active)
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (3)
      203.0.113.1     209.165.201.1    10.0.0.1   198.51.100.1
      172.16.0.1
    IPv6: (0)
  Peer holdtime: 180 sec; KA interval: 60 sec; Peer state: Estab
  NSR: Disabled
  Clients: AToM
  Capabilities:
    Sent:
      0x508  (MP: Point-to-Multipoint (P2MP))
      0x509  (MP: Multipoint-to-Multipoint (MP2MP))
      0x50b  (Typed Wildcard FEC)
      0x50d  (State Advertisement Control)
        [ {IPv4-disable}  ] (length 1)
    Received:
      0x508  (MP: Point-to-Multipoint (P2MP))
      0x509  (MP: Multipoint-to-Multipoint (MP2MP))
      0x50b  (Typed Wildcard FEC)
      0x50d  (State Advertisement Control)
```

Capabilities Sent SAC capability **ipv4-disable** is sent, and local IPv4 label bindings are not generated.

Capabilities Received The peer (PE2) understands SAC capability and won't send its local IPv4 label bindings to local PE.

On PE1, verify SAC capabilities:

```
PE1# show mpls ldp capabilities detail

  Type     Description                                Owner
  ------   ----------------------------------------   ------------
```

```
0x50b    Typed Wildcard FEC                          LDP
   Capability data: None

0x3eff   Cisco IOS-XR                                LDP
   Capability data:
     Length: 12
     Desc  : [ host=PE1; platform=ASR9000; release=07.01.01 ]

0x508    MP: Point-to-Multipoint (P2MP)              mLDP
   Capability data: None

0x509    MP: Multipoint-to-Multipoint (MP2MP)        mLDP
   Capability data: None

0x50d    State Advertisement Control                 LDP
   Capability data:
     Length: 1
     Desc  : [ {IPv4-disable}  ]

0x703    P2MP PW                                     L2VPN-AToM
   Capability data: None
```

On PE1, verify that local and remote FEC bindings are removed.

```
PE1# show mpls ldp neighbor 198.51.100.1
Wed March 3 13:42:13.359 EDTs
```

ECMP and Bundle Hashing with Entropy Label

Entropy label (EL) improves load balancing across a network. Load balancing helps in planning the capacity of a network by distributing traffic across multiple paths that are based on hashing functions.

Note The routers do not support imposition or disposition of EL.

Traffic load balancing over Equal Cost Multipath (ECMP) or Link Aggregation Groups (LAGs) is based on a hashing function. To arrive at the hash calculations, the node that performs the load balancing must read the header fields in the incoming packets. Currently, Label Switching Routers (LSRs) at each transit point must do a Deep Packet Inspection (DPI) along the path of a given Label Switched Path (LSP). This includes extracting the appropriate keys for load balancing. If the LSR is unable to infer the protocol, it uses the topmost MPLS labels in the label stack as keys to balance load. This may result in an unbalanced distribution of traffic.

Entropy labels enhance load balancing by eliminating the need for DPI at the transit LSRs. The transit router recognizes the incoming MPLS packets with an entropy label and performs the load balancing and forwards the MPLS packet on a selected path.

The ingress LSR of an LSP computes the hash that is based on appropriate fields from a given packet and places the result in a label that is called an entropy label as part of the MPLS label stack. Using the entropy label in the hash keys reduces the need of a DPI in the LSR. The transit LSR can use the entire label stack of the MPLS packet to perform load balancing, as the entropy label introduces the right level of order into the label stack.

For more information on EL, see *RFC 6790*.

MPLS Hashing

The hashing uses the label stack and the payload when the label stack contains EL. However, load balancing functionality considers EL like any other label.

Starting from the Cisco IOS-XR Release 7.3.1, hashing is performed as described in the following table:

Table 4: MPLS Hashing

MPLS Payload	Fields Considered for Hashing	Description
IPv4	Router ID + Label stack + Src IP + Dst IP + L4 Protocol + Src Port + Dst Port	If IPv4 is the MPLS payload, hashing uses: • Up to 14 labels in the label stack. • Source and destination ports are used if the protocol is UDP or TCP. The cyclic redundancy check (CRC) hash value(s) for the same packet varies based on whether the ingress network processing unit (NPU) is Q100 or Q200.
IPv6	Router ID + Label stack + Src IP + Dst IP + Flow-label + L4 Protocol + Src Port + Dst Port	If IPv6 is the MPLS payload, hashing uses: • Up to 14 labels in the label stack. • Source and destination ports are used if the protocol is UDP or TCP. The cyclic redundancy check (CRC) hash value(s) for the same packet varies based on whether the ingress network processing unit (NPU) is Q100 or Q200.
GTP-u (IPv4 or IPv6)	Router ID + Label stack + GTP TEID	GPRS tuneling protocol. GTP-u uses UDP destination port 2152. The same fields are used for both IPv4 and IPv6. GTP TEID is a 32-bit tunnel end-point identifier.

MPLS Payload	Fields Considered for Hashing	Description
Ethernet	Router ID + Label stack + Dest MAC + Src MAC + Ether-type + VLAN	If Ethernet is the MPLS payload, hashing uses: • Up to 14 labels in the label stack. • For untagged Ethernet, destination MAC address, source MAC address, and Ethernet type are used. • For tagged Ethernet, destination MAC address, source MAC address, Ethernet type, and first VLAN tags are used. • When L2VPN is configured with the Control word, destination MAC address and source MAC address are used. The cyclic redundancy check (CRC) hash value(s) for the same packet varies based on whether the ingress network processing unit (NPU) is Q100 or Q200.

Load Balancing based on the Position of Entropy Label

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
Load Balancing based on the Position of Entropy Label	Release 7.5.2	The router achieves multi-pathing or traffic load balancing based on the Entropy Label (EL) and Entropy Label Indicator (ELI) position. The load balancing is based on the router ID and label stack if the ELI is in any of the three bottom labels. If not, the load balancing is based on all labels in the label stack plus the MPLS payload.

MPLS label stack contains Entropy Label (EL) and Entropy Label Indicator (ELI). The label immediately preceding an EL in the MPLS label stack is an ELI. The ELI uses a reserved label value of 7.

Starting from the Cisco IOS-XR Release 7.5.2, the load balancing is based on the placement of the EL and ELI. If the ELI and EL are placed at the bottom three label entry positions, load balancing uses all labels, else uses the MPLS payload along with other labels.

For GTP payload, the presence of EL does not change the hashing mechanism and it remains the same as described in the *MPLS Hashing* table.

The following table shows how load balancing is performed based on the position of ELI.

Position of ELI (Label Value 7)	Description
Bottom three labels	If there is ELI in any of the three bottom label positions, the load balacing is based on the router ID and label stack. The load balacing uses up to 14 labels in the label stack.
Anywhere else in the stack	The load balacing is performed based on the entropy label as described in the *MPLS Hashing* table. The load balacing uses all labels in the label stack and MPLS payload.

Additional References

For additional information related to Implementing MPLS Label Distribution Protocol, refer to the following references:

Related Documents

Related Topic	Document Title
LDP Commands	*MPLS Label Distribution Protocol Commands* module in *MPLS Command Reference for Cisco 8000 Series Routers*.

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

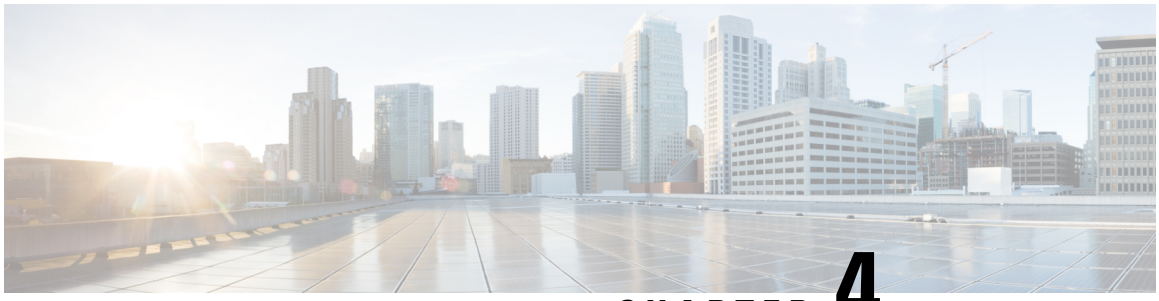
MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs **Note** Not all supported RFCs are listed.	Title
RFC 3031	*Multiprotocol Label Switching Architecture*
RFC 3036	*LDP Specification*
RFC 3037	*LDP Applicability*
RFC 3478	*Graceful Restart Mechanism for Label Distribution Protocol*
RFC 3815	*Definitions of Managed Objects for MPLS LDP*
RFC 5036	*Label Distribution and Management* *Downstream on Demand Label Advertisement*
RFC 5286	*Basic Specification for IP Fast Reroute: Loop-Free Alternates*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport



CHAPTER 4

Implementing MPLS OAM

IP-Less MPLS-TP Ping and MPLS-TP Traceroute

In Label Switched Path (LSP) ping or traceroute with IP encapsulation over ACH, IP encapsulated ping or traceroute packets are sent over the MPLS LSP using the control channel (ACH). The application-level control channel in this case is the reverse path of the LSP using ACH. The on-demand ping or traceroute echo response message is sent on the reverse path of the LSP. The response uses ACH and is IP encapsulated. The destination address in the IP header is set to that of the sender of the echo request message, and the source address in the IP header is set to a valid address of the replying node.

- the reply mode is 4
- the node does not have a return MPLS LSP path to the echo request source.

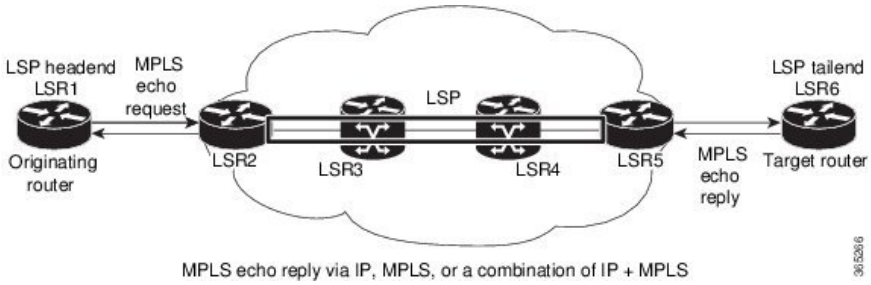
MPLS LSP Ping

The MPLS LSP Ping feature is used to check the connectivity between Ingress LSR and egress LSRs along an LSP. MPLS LSP ping uses MPLS echo request and reply messages, similar to Internet Control Message Protocol (ICMP) echo request and reply messages, to validate an LSP. While ICMP echo request and reply messages validate IP networks, MPLS echo and reply messages validate MPLS networks. The MPLS echo request packet is sent to a target router through the use of the appropriate label stack associated with the LSP to be validated. Use of the label stack causes the packet to be forwarded over the LSP itself. The destination IP address of the MPLS echo request packet is different from the address used to select the label stack. The destination IP address is defined as a 127.x.y.z/8 address and it prevents the IP packet from being IP switched to its destination, if the LSP is broken.

An MPLS echo reply is sent in response to an MPLS echo request. The reply is sent as an IP packet and it is forwarded using IP, MPLS, or a combination of both types of switching. The source address of the MPLS echo reply packet is an address obtained from the router generating the echo reply. The destination address is the source address of the router that originated the MPLS echo request packet. The MPLS echo reply destination port is set to the echo request source port.

The following figure shows MPLS LSP ping echo request and echo reply paths.

Figure 9: MPLS LSP Ping Echo Request and Reply Paths



By default, the **ping mpls ipv4** command tries to determine the Forwarding Equivalence Class (FEC) being used automatically. However, this is only applicable at head-end and works only if the FEC at the destination is same as the source. If the source and destination FEC types are not the same, the **ping mpls ipv4** command may fail to identify the targeted FEC type. You can overcome this limitation by specifying the FEC type in MPLS LSP ping using the **fec-type** command option. If the user is not sure about the FEC type at the transit or the destination, or it may change through network, use of the **generic** FEC type command option is recommended. Generic FEC is not coupled to a particular control plane and allows path verification when the advertising protocol is unknown, or may change during the path of the echo request. If you are aware of the destination FEC type, specify the target FEC as BGP or LDP.

Configuration Examples

This example shows how to use MPLS LSP ping to test the connectivity of an IPv4 LDP LSP. The destination is specified as a Label Distribution Protocol (LDP) IPv4 address.

```
RP/0/RP0/CPU0:router# ping mpls ipv4 10.1.1.2/32 verbose

Sun Nov 15 11:27:43.070 UTC

Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
      timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

!      size 100, reply addr 10.1.0.2, return code 3
!      size 100, reply addr 10.1.0.2, return code 3
!      size 100, reply addr 10.1.0.2, return code 3
!      size 100, reply addr 10.1.0.2, return code 3
!      size 100, reply addr 10.1.0.2, return code 3

Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/4 ms
```

In this example, the destination is specified as a Label Distribution Protocol (LDP) IPv4 prefix and Forwarding Equivalence Class (FEC) type is specified as generic.

```
RP/0/RP0/CPU0:router# ping mpls ipv4 10.1.1.2/32 fec-type generic

Wed Nov 25 03:36:33.143 UTC
```

```
Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
      timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

In this example, the destination is specified as a Label Distribution Protocol (LDP) IPv4 prefix and the FEC type is specified as BGP.

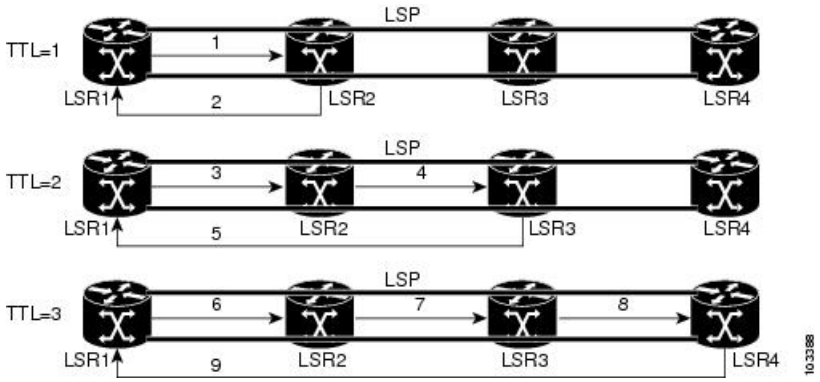
```
RP/0/RP0/CPU0:router# ping mpls ipv4 10.1.1.2/32 fec-type bgp

Wed Nov 25 03:38:33.143 UTC
Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
      timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

MPLS LSP Traceroute

The MPLS LSP Traceroute feature is used to isolate the failure point of an LSP. It is used for hop-by-hop fault localization and path tracing. The MPLS LSP Traceroute feature relies on the expiration of the Time to Live (TTL) value of the packet that carries the echo request. When the MPLS echo request message hits a transit node, it checks the TTL value and if it is expired, the packet is passed to the control plane, else the message is forwarded. If the echo message is passed to the control plane, a reply message is generated based on the contents of the request message.

The following figure shows an MPLS LSP traceroute example with an LSP from LSR1 to LSR4.

Figure 10: MPLS LSP Traceroute



By default, the **traceroute mpls ipv4** command tries to determine the Forwarding Equivalence Class (FEC) being used automatically. However, this is only applicable at head-end and works only if the FEC at the destination is same as the source. If the source and destination FEC types are not the same, the **traceroute mpls ipv4** command may fail to identify the targeted FEC type. You can overcome this limitation by specifying the FEC type in MPLS LSP traceroute using the **fec-type** command option. If the user is not sure about the FEC type at the transit or the destination, or it may change through network, use of the **generic** FEC type command option is recommended. Generic FEC is not coupled to a particular control plane and allows path verification when the advertising protocol is unknown, or may change during the path of the echo request. If you are aware of the destination FEC type, specify the target FEC as BGP or LDP.

Configuration Examples

This example shows how to use the **traceroute** command to trace to a destination.

```
RP/0/RP0/CPU0:router# traceroute mpls ipv4 10.1.1.2/32 destination 127.0.0.3 127.0.0.6 2
Sat Jan 27 03:50:23.746 UTC

Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

Destination address 127.0.0.3
  0 10.2.1.2 MRU 1500 [Labels: 24000 Exp: 0]
L 1 10.2.1.1 MRU 1500 [Labels: implicit-null Exp: 0] 8 ms
! 2 10.1.0.2 3 ms

Destination address 127.0.0.5
  0 10.2.1.2 MRU 1500 [Labels: 24000 Exp: 0]
L 1 10.2.1.1 MRU 1500 [Labels: implicit-null Exp: 0] 5 ms
! 2 10.1.0.2 2 ms
```

This example shows how to use the **traceroute** command and how to specify the maximum number of hops for the traceroute to traverse by specifying the **ttl** value.

```
RP/0/RP0/CPU0:router# traceroute mpls ipv4 10.1.1.2/32 ttl 1
Sun Nov 15 12:20:14.145 UTC
Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

  0 10.1.0.1 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 10.1.0.2 3 ms
```

This example shows how to use the **traceroute** command to trace to a destination and FEC type is specified as generic.

```
RP/0/RP0/CPU0:router# traceroute mpls ipv4 10.1.1.2/32 fec-type generic
Sun Nov 15 12:25:14.145 UTC
Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'L' - labeled output interface, 'B' - unlabeled output interface,
'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
'P' - no rx intf label prot, 'p' - premature termination of LSP,
'R' - transit router, 'I' - unknown upstream index,
'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
0 10.12.12.1 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 10.12.12.2 2 ms
```

This example shows how to use the **traceroute** command to trace to a destination and FEC type is specified as BGP.

```
RP/0/RP0/CPU0:router# traceroute mpls ipv4 10.1.1.2/32 fec-type bgp
Sun Nov 15 12:25:14.145 UTC
Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'L' - labeled output interface, 'B' - unlabeled output interface,
'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
'P' - no rx intf label prot, 'p' - premature termination of LSP,
'R' - transit router, 'I' - unknown upstream index,
'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
0 10.12.12.1 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 10.12.12.2 2 ms
```

MPLS OAM Using Nil FEC

The Nil-FEC LSP ping and traceroute operations are extensions of regular MPLS ping and traceroute. MPLS ping and traceroute requires at least one forwarding equivalence class (FEC) in the target FEC stack. In Nil-FEC ping and traceroute operations, an explicit FEC is not associated with the label. Nil-FEC LSP ping and traceroute support MPLS static LSPs and also act as an additional diagnostic tool for all other LSP types.

Nil-FEC LSP ping and traceroute allow network operators to provide the ability to freely test any label stack by allowing them to specify the following:

- label stack
- outgoing interface
- nexthop address

The following table shows the syntax for the ping and traceroute commands.

Table 6: LSP Ping and Traceroute Nil FEC Commands

Command Syntax
ping mpls nil-fec labels {*label*[**,***label*]} [**output** {**interface** *tx-interface*} [**nexthop** *nexthop-ip-addr*]]
traceroute mpls nil-fec labels {*label*[**,***label*]} [**output** {**interface** *tx-interface*} [**nexthop** *nexthop-ip-addr*]]

Examples: LSP Ping Nil FEC and LSP Traceroute Nil FEC

The examples in this section use the following topology:

```
Node loopback IP address: 172.18.1.3   172.18.1.4   172.18.1.5   172.18.1.7
Node label:               16003        16004        16005        16007
Nodes:                    Arizona ---- Utah ------- Wyoming ---- Texas

Interface:            GigabitEthernet0/2/0/1   GigabitEthernet0/2/0/1
Interface IP address:         10.1.1.3              10.1.1.4
```

```
RP/0/RP0/CPU0:router-arizona# show mpls forwarding

Tue May  2 13:44:31.999 EDT
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
16004  Pop         No ID             Gi0/2/0/1    10.1.1.4        1392
       Pop         No ID             Gi0/2/0/2    10.1.2.2        0
16005  16005       No ID             Gi0/2/0/0    10.1.1.4        0
       16005       No ID             Gi0/2/0/1    10.1.2.2        0
16007  16007       No ID             Gi0/2/0/0    10.1.1.4        4752
       16007       No ID             Gi0/2/0/1    10.1.2.2        0
```

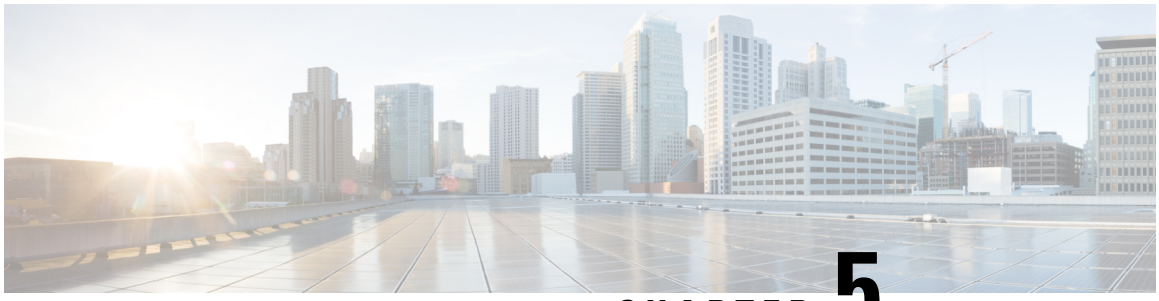
This example shows how to use Nil-FEC LSP ping to test a label stack.

```
RP/0/RP0/CPU0:router-arizona# ping mpls nil-fec labels 16005,16007 output interface
GigabitEthernet 0/2/0/1 nexthop 10.1.1.4 repeat 1
Sending 1, 72-byte MPLS Echos with Nil FEC labels 16005,16007,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
```

```
  'd' - see DDMAP for return code,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 1/1/1 ms
 Total Time Elapsed 0 ms
```

This example shows how to use Nil-FEC LSP traceroute for a label stack.

```
RP/0/RP0/CPU0:router-arizona# traceroute mpls nil-fec labels 16005,16007 output interface
GigabitEthernet 0/2/0/1 nexthop 10.1.1.4
Tracing MPLS Label Switched Path with Nil FEC labels 16005,16007, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'd' - see DDMAP for return code,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
  0 10.1.1.3 MRU 1500 [Labels: 16005/16007/explicit-null Exp: 0/0/0]
L 1 10.1.1.4 MRU 1500 [Labels: implicit-null/16007/explicit-null Exp: 0/0/0] 1 ms
L 2 10.1.1.5 MRU 1500 [Labels: implicit-null/explicit-null Exp: 0/0] 1 ms
! 3 10.1.1.7 1 ms
```



CHAPTER 5

MPLS Static Labeling

The MPLS static feature enables you to statically assign local labels to an IPv4/Ipv6 prefix. Also, Label Switched Paths (LSPs) can be provisioned for these static labels by specifying the next-hop information that is required to forward the packets containing static label.

If there is any discrepancy between labels assigned statically and dynamically, the router issues a warning message in the console log. By means of this warning message, the discrepancy can be identified and resolved.

The advantages of static labels over dynamic labels are:

- Improves security because the risk of receiving unwanted labels from peers (running a compromised MPLS dynamic labeling protocol) is reduced.
- Gives users full control over defined LSPs.
- Utilize system resources optimally because dynamic labeling is not processed.

Restrictions

- Static labeling on IPv6 packets is not supported.
- The router does not prevent label discrepancy at the time of configuring static labels. Any generated discrepancy needs to be subsequently cleared.
- Equal-cost multi-path routing (ECMP) is not supported.
- Interfaces must be explicitly configured to handle traffic with static MPLS labels.
- The MPLS per-VRF labels cannot be shared between MPLS static and other applications.

Forwarding Labeled Packets

This section describes how labeled packets are forwarded in MPLS networks, how forwarding labeled packets are different from forwarding IP packets, how labeled packets are load-balanced, and what a LSR does with a packet with an unknown label.

Top Label Value

When a labeled packet is received, the label value at the top of the stack is looked up. The LSR sees the 20-bit field in the top label, which carries the actual value of the label. As a result of a successful lookup, the LSR learns:

- the next hop to which the packet is to be forwarded.
- what label operation to be performed before forwarding - swap, push, or pop.

The processing is always based on the top label, without regard to the possibility that in the past some other number of another label may have been "above it", or at present that some other number of another label may be below it. An unlabeled packet can be thought of as a packet whose label stack is empty (that is, a packet whose label stack has depth zero).

IP Lookup Versus Label Lookup

When a router receives an IP packet, an IP lookup is done. This means that the packet is looked up in the Cisco Express Forwarding (CEF) table. When a router receives a labeled packet, the label forwarding information base (LFIB) of the router is looked up. The router knows by looking at the protocol field in the Layer 2 header what type of packet it receives: a labeled packet or an IP packet.

Load Balancing Labeled Packets

If multiple equal-cost paths exist for an IPv4 prefix, Cisco IOS XR Software can load-balance labeled packets. When labeled packets are load-balanced, they can have the same or different outgoing labels. The outgoing labels are the same if the two links are between a pair of routers and both links belong to the platform label space. If multiple next-hop LSRs exist, the outgoing label for each path is usually different, because the next-hop LSRs assign labels independently.

Unknown Label

In regular operations, an LSR should receive only a labeled packet with a label at the top of the stack that is known to the LSR, because the LSR would have previously advertised that label. However, it is possible, in some cases, when something goes amiss in the MPLS network, the LSR starts receiving labeled packets with a top label that the LSR does not find in its LFIB. In such cases, the LSR drops the packet.

Define Label Range and Enable MPLS Encapsulation

By default, MPLS encapsulation is disabled on all interfaces. MPLS encapsulation has to be explicitly enabled on all ingress and egress MPLS interfaces through which the static MPLS labeled traffic travels.

Also, the dynamic label range needs to be defined. Any label that falls outside this dynamic range is available for manually allocating as static labels. The router does not verify statically-configured labels against the specified label range. Therefore, to prevent label discrepancy, ensure that you do not configure static MPLS labels that fall within the dynamic label range.

Note For Cisco IOS XR software release 7.5.2 onwards, MPLS static supports 200G Ethernet.

Configuration Example

You have to accomplish the following to complete the MPLS static labeling configuration. Values are provided as an example.

1. Define a dynamic label range, which in this task is set between 17000 and 18000.
2. Enable MPLS encapsulation on the required interface.
3. Setup a static MPLS LSP for a specific ingress label 24035.
4. Specify the forwarding information so that for packets that are received with the label, 24035, the MPLS protocol swaps labels and applies the label, 24036. After applying the new label, it forwards the packets to the next hop, 10.2.2.2, through the specified interface.

```
Router(config)#mpls label range table 0 17000 18000
Router(config)#commit
Router(config)#mpls static
Router(config-mpls-static)#interface HundredGigE 0/0/0/25
Router(config-mpls-static)#address-family ipv4 unicast
Router(config-mpls-static-af)#local-label 24035 allocate
Router(config-mpls-static-af-lbl)#forward
Router(config-mpls-static-af-lbl-fwd)#path 1 nexthop HundredGigE 0/0/0/27 10.2.2.2 out-label
 24036
Router(config-mpls-static-af-lbl-fwd)# commit
```

Verification

Verify the interfaces on which MPLS is enabled

```
Router# show  mpls interfaces

Interface                  LDP      Tunnel   Static   Enabled
-------------------------- -------- -------- -------- --------
HundredGigE 0/0/0/25             No       No       Yes     Yes
```

Verify that the status is "Created" for the specified label value.

```
Router#show mpls static local-label all

Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
24035   default         X-Connect    NA               Yes             Created
```

Check the dynamic range and ensure that the specified local-label value is outside this range.

```
Router#show mpls label range

Range for dynamic labels: Min/Max: 17000/18000
```

Verify that the MPLS static configuration has taken effect, and the label forwarding is taking place.

```
Router#show mpls lsd forwarding

In_Label, (ID), Path_Info: <Type>
```

```
24035, (Static), 1 Paths
   1/1: IPv4, 'default':4U, BE1.2, nh=10.20.3.1, lbl=35001, flags=0x0, ext_flags=0x0
```

Associated Commands

- mpls static
- mpls label range
- show mpls interfaces

Identify and Clear Label Discrepancy

During configuring or de-configuring static labels or a label range, a label discrepancy can get generated when:

- A static label is configured for an IP prefix that already has a binding with a dynamic label.
- A static label is configured for an IP prefix, when the same label value is dynamically allocated to another IP prefix.

Verification

Identify label discrepancy by using these show commands.

```
Router#show mpls static local-label discrepancy
Tue Apr 22 18:36:31.614 UTC
Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
24000   default         X-Connect     NA              Yes             Discrepancy


Router#show mpls static local-label all
Tue Apr 22 18:36:31.614 UTC
Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
24000   default         X-Connect    N/A              Yes             Discrepancy
24035   default         X-Connect    N/A              Yes             Created


Router#show log

Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: level warnings, 199 messages logged
    Monitor logging: level debugging, 0 messages logged
    Trap logging: level informational, 0 messages logged
    Buffer logging: level debugging, 2 messages logged

Log Buffer (307200 bytes):

RP/0/RSP0/CPU0:Apr 24 14:18:53.743 : mpls_static[1043]:
%ROUTING-MPLS_STATIC-7-ERR_STATIC_LABEL_DISCREPANCY :
The system detected 1 label discrepancies (static label could not be allocated due to
conflict with other applications).
Please use 'clear mpls static local-label discrepancy' to fix this issue.
RP/0/RSP0/CPU0:Apr 24 14:18:53.937 : config[65762]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
 committed by user 'cisco'.
Use 'show configuration commit changes 1000000020' to view the changes.
```

Rectification

Label discrepancy is cleared by allocating a new label to those IP prefixes that are allocated dynamic label. The static label configuration takes precedence while clearing discrepancy. Traffic can be affected while clearing discrepancy.

```
Router# clear mpls static local-label discrepancy all
```

Verify that the discrepancy is cleared.

```
Router# show mpls static local-label all
Wed Nov 25 21:45:50.368 UTC
Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
24000   default         X-Connect    N/A              Yes             Created
24035   default         X-Connect    N/A              Yes             Created
```

Associated Commands

- show mpls static local-label discrepancy
- clear mpls static local-label discrepancy all

Configuring Backup within a Forwarding Set

Various types of FRR backups can be configured between links with in a forwarding path set. You can configure the following types of FRR backups:

- Pure FRR Backup
- Reciprocal FRR backup
- One-way FRR backup

In pure FRR backup, there will be separate primary paths and backup paths. In reciprocal FRR backup, each path can act as both primary and backup. In one-way FRR backup, some paths act as both primary and backup while other paths may be just primary paths or backup paths.

Configuration Example: Pure FRR Backup

This example shows how to configure pure FRR backup with in a forwarding path set.

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# lsp lsp1
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 25000 allocate
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 nexthop hundredGigE 0/0/0/25 10.1.0.1
 out-label 25000 backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 2 nexthop hundredGigE 0/0/0/26 10.1.0.3
 out-label 25001 backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# exit
RP/0/0/CPU0:Router(config-mpls-static-lsp)# backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 1 nexthop hundredGigE 0/0/0/27
10.5.0.1 out-label pop backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 2 nexthop hundredGigE 0/0/0/28
10.6.0.2 out-label pop backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# exit
```

The following table describes the forwarding behavior for pure FRR backup. Here P1-F and P2-F are the forwarding paths and P1-B and P2-B are the backup paths.

Action	Transient State	Interface Steady State	Forward Steady State
N/A	N/A	• P1-F: Up P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Backup • P1-B: N/A P2-B: N/A
P1-F Down	P1-F FRR to P2-F	• P1-F: Up P2-F: Down • P1-B: Up P2-B: Up	• P1-F: Down P2-F: Flow • P1-B: Backup P2-B: N/A
P2-F Down	P2-F FRR to P1-B	• P1-F: Down P2-F: Down • P1-B: Up P2-B: Up	• P1-F: Down P2-F: Down • P1-B: Flow P2-B: Backup
P1-B Down	P1-B FRR to P2-B	• P1-F: Down P2-F: Down • P1-B: Down P2-B: Up	• P1-F: Down P2-F: Down • P1-B: Down P2-B: Flow

Configuration Example: Reciprocal FRR Backup

This example shows how to configure reciprocal FRR backup with in a forwarding path set.

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# lsp lsp1
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 25000 allocate
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 nexthop hundredGigE 0/0/0/25 10.1.0.1
 out-label 25000 primary-and-backup backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 2 nexthop hundredGigE 0/0/0/26 10.1.0.3
 out-label 25001 primary-and-backup backup-id 1
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# exit
RP/0/0/CPU0:Router(config-mpls-static-lsp)# backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 1 nexthop hundredGigE 0/0/0/27
10.5.0.1 out-label pop primary-and-backup backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 2 nexthop hundredGigE 0/0/0/28
10.6.0.2 out-label pop primary-and-backup backup-id 1
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# exit
```

The following table describes the forwarding behavior for reciprocal FRR backup.

Action	Transient State	Interface Steady State	Forward Steady State
N/A	N/A	• P1-F: Up P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Flow • P1-B: N/A P2-B: N/A
P2-F Down	P2-F FRR to P1-F	• P1-F: Down P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Down • P1-B: Backup P2-B: N/A
P1-F Down	P1-F FRR to P1-B	• P1-F: Down P2-F: Down • P1-B: Up P2-B: Up	• P1-F: Down P2-F: Down • P1-B: Flow P2-B: Flow
P2-B Down	P2-B FRR to P1-B	• P1-F: Down P2-F: Down • P1-B: Up P2-B: Down	• P1-F: Down P2-F: Down • P1-B: Flow P2-B: Down

Configuration Example: One-way FRR Backup

This example shows how to configure one-way FRR backup with in a forwarding path set.

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# lsp lsp1
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 25000 allocate
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 nexthop hundredGigE 0/0/0/25 10.1.0.1
 out-label 25000 backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 2 nexthop hundredGigE 0/0/0/26 10.1.0.3
 out-label 25001 primary-and-backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# exit
RP/0/0/CPU0:Router(config-mpls-static-lsp)# backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 1 nexthop hundredGigE 0/0/0/27
10.5.0.1 out-label pop backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 2 nexthop hundredGigE 0/0/0/28
10.6.0.2 out-label pop primary-and-backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# exit
```

The following table describes the forwarding behavior for one-way FRR backup.

Action	Transient State	Interface Steady State	Forward Steady State
N/A	N/A	• P1-F: Up P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Flow • P1-B: N/A P2-B: N/A
P2-F Down	P2-F NO-FRR to P1-F	• P1-F: Down P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Down • P1-B: Backup P2-B: N/A
P1-F Down	P1-F FRR to P1-B	• P1-F: Down P2-F: Down • P1-B: Up P2-B: Up	• P1-F: Down P2-F: Down • P1-B: Flow P2-B: Flow
P1-B Down	P1-B FRR to P2-B	• P1-F: Down P2-F: Down • P1-B: Down P2-B: Up	• P1-F: Down P2-F: Down • P1-B: Down P2-B: Flow

Configuring Static LSP Next Hop Resolve

You can specify the outgoing next hop instead of explicitly specifying the outgoing path while configuring static LSPs. This next hop is resolved using the routing information base (RIB) which provides a list of paths to auto-configure. While specifying the next hop for the incoming label in a static LSP, you can specify the next hop address with out the interface using the **resolve-nexthop** command.

The following restrictions apply for this feature:

- Only supports a single next hop address which may resolve to multiple paths.
- Non-default VRFs are not supported.

Configuration Example

This example shows how to configure the static LSP next hop without specifying the interface using the **resolve-nexthop** command.

```
Router# configure terminal
Router(config)# mpls static
Router(config-mpls-static)# lsp ipv6-2
Router(config-mpls-static-lsp)# in-label 25000 allocate per-prefix 2001:DB8:0:1::/64  or
24:24:1::/64
Router(config-mpls-static-lsp)# forward
Router(config-mpls-static-lsp-fwd)# path 1 resolve-nexthop 2001:DB8:0:2::64 out-label pop
Router(config-mpls-static-lsp-fwd)# exit
```

Configuring Static LSP Next Hop Resolve with Recursive Prefix

When a routing table entry references to another IP address and not to a directly connected exit interface, the next-hop IP address is resolved using another route with an exit interface. This is known as a recursive look up because multiple lookups are required to resolve the next-hop IP address. Static LSP next hop resolve with recursive prefix feature supports resolution of recursive routes for static LSPs. In this feature, you can specify a next hop which is not directly connected using the **resolve-nexthop** command for a static LSP.

Restrictions

The following restrictions apply for this feature:

- Only eBGP routes are supported.

Configuration Example

This example shows how to configure the static LSP next hop resolve with recursive prefix. Here 192.168.2.1 is a recursive route learnt through eBGP.

```
Router# configure terminal
Router(config)# mpls static
Router(config-mpls-static)# lsp anycast_5001
Router(config-mpls-static-lsp)# in-label 5001 allocate
Router(config-mpls-static-lsp)# forward
Router(config-mpls-static-lsp-fwd)# path 1 resolve-nexthop 192.168.2.1 out-label pop
Router(config-mpls-static-lsp-fwd)# exit
```

Verification

This example shows how to verify the static LSP next hop resolve with recursive prefix configuration.

```
Router# show  mpls static lsp anycast_5001 detail

LSP Name             Label   VRF             AFI  Type         Prefix           RW Configured
   Status
-------------------- ------- --------------- ---- ------------ -----------------
--------------- --------
anycast_5001         5001    default         N/A  X-Connect    N/A              Yes
      Created
  PRIMARY SET:
    [resolve-mode: nexthop 192.168.2.1]
    Path 0 : nexthop BVI1 10.1.1.3, out-label Pop, Role: primary, Path-id: 0, Status: valid

    Path 1 : nexthop BVI1 10.1.1.4, out-label Pop, Role: primary, Path-id: 0, Status: valid

    Path 2 : nexthop BVI1 10.1.1.5, out-label Pop, Role: primary, Path-id: 0, Status: valid

    Path 3 : nexthop BVI1 10.1.1.6, out-label Pop, Role: primary, Path-id: 0, Status: valid
```

MPLS Static Labeling

The MPLS static feature enables you to statically assign local labels to an IPv4/Ipv6 prefix. Also, Label Switched Paths (LSPs) can be provisioned for these static labels by specifying the next-hop information that is required to forward the packets containing static label.

If there is any discrepancy between labels assigned statically and dynamically, the router issues a warning message in the console log. By means of this warning message, the discrepancy can be identified and resolved.

The advantages of static labels over dynamic labels are:

- Improves security because the risk of receiving unwanted labels from peers (running a compromised MPLS dynamic labeling protocol) is reduced.
- Gives users full control over defined LSPs.
- Utilize system resources optimally because dynamic labeling is not processed.

Restrictions

- Static labeling on IPv6 packets is not supported.
- The router does not prevent label discrepancy at the time of configuring static labels. Any generated discrepancy needs to be subsequently cleared.
- Equal-cost multi-path routing (ECMP) is not supported.
- Interfaces must be explicitly configured to handle traffic with static MPLS labels.
- The MPLS per-VRF labels cannot be shared between MPLS static and other applications.

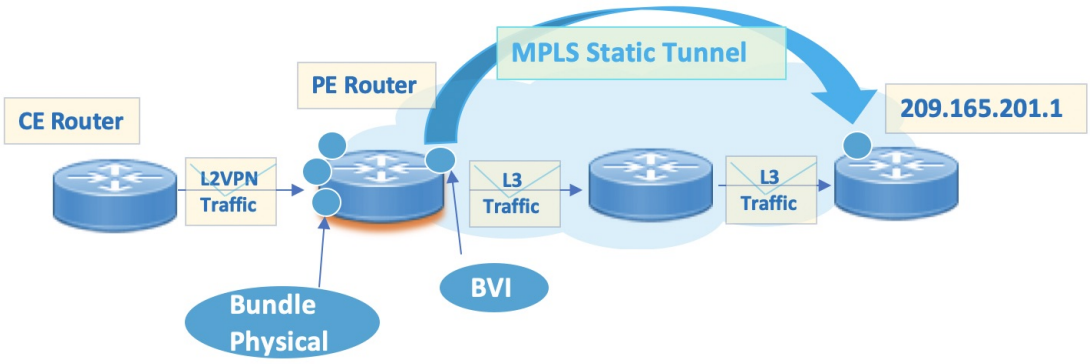
MPLS Static Forwarding Over A BVI

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
MPLS Static Forwarding Over A BVI	Release 7.3.1	The router can receive MPLS L2VPN traffic from an L2 bridge domain, and forward the L3 (customer) traffic over an egress BVI, using an MPLS static LSP. For the incoming L2VPN traffic, the BVI serves as an L3 gateway. Since the router can perform switching for L2 traffic and routing for incoming L3 MPLS traffic, it enhances flexibility for transporting MPLS traffic.

Consider this sample topology, connecting a CE router to a PE router.

Figure 11: MPLS Static Forwarding Over A BVI



Pointers

- L2VPN packets are attached to specific bridge domains, and correspond to a VLAN (or 802.1Q tag). In turn, the VLANs are associated with specific bundle or physical interfaces for sending traffic between the CE and PE routers. These associations have to be configured on the CE and PE routers for transporting L2VPN traffic.
- The L2VPN traffic encapsulates (IPv4 or IPv6) customer payload, and is sent from the CE router to the PE router.
- The PE router does an MPLS label lookup on the incoming MPLS traffic, and removes the VLAN (or 802.1Q) header. In general, the router can perform a label operation like swap, PHP, or pop. After removing the VLAN header, the (previously) encapsulated IP traffic is sent towards the bridge-group virtual interface (BVI).
- With BVI support for the MPLS static function, the incoming labelled traffic can be resolved using a static LSP. The BVI resolves the nexthop to an L3 interface.

BVI pointers

- A BVI next hop can be a static route, a directly connected route, or a route resolved through BGP or an IGP.
- Only an MPLS static LSP can use a BVI as a next hop.

Configuration

The configurations explain how to enable forwarding of (incoming) L2VPN traffic over a (outgoing) BVI, through an MPLS static LSP.

Interfaces Configuration

- The **l2transport** keyword indicates that the interface is an L2 interface, and the L2 traffic belongs to the VLANs specified in the dot1q tags.
- The **rewrite ingress tag pop** command form instructs the router to remove the 802.1Q (or VLAN) tag and forward the payload.
- Corresponding configurations should also be enabled on the CE router.

```
Router# configure terminal
Router(config)# interface Bundle-Ether101.101 l2transport
Router(config-if)# encapsulation dot1q 2001
Router(config-if)# rewrite ingress tag pop 1 symmetric
Router(config-if)# mtu 2000

Router# configure terminal
Router(config)# interface Bundle-Ether102.101 l2transport
Router(config-if)# encapsulation dot1q 3001
Router(config-if)# rewrite ingress tag pop 1 symmetric
Router(config-if)# mtu 2000

Router# configure terminal
Router(config)# interface HundredGigE0/11/0/25.101 l2transport
Router(config-if)# encapsulation dot1q 101
Router(config-if)# rewrite ingress tag pop 1 symmetric

Router# configure terminal
Router(config)# interface HundredGigE0/11/0/31.101 l2transport
Router(config-if)# encapsulation dot1q 1001
Router(config-if)# rewrite ingress tag pop 1 symmetric
Router(config-if)# exit
Router(config)# commit
```

BVI Configuration

The BVI acts as an L3 gateway for the ingress L2VPN traffic. The customer IP traffic is sent to the BVI IP address that is configured in this task.

```
Router# configure terminal
Router(config)# interface BVI101
Router(config-if)# ipv4 address 192.0.2.1 255.255.255.0
Router(config-if)# ipv6 address 2001:DB8:A:B::1/64
Router(config-if)# mac-address 0.10.5500
Router(config-if)# load-interval 30
Router(config-if)# commit
```

L2VPN Configuration

- L2VPN is configured, and a bridge domain (bd1) is associated with it.
- A bundle interface and BVI are associated with the BD. The instructions enable VLAN traffic received at the bundle interface (Bundle-Ether101.101) to be routed through the L3 gateway, BVI BVI101. VLAN-to-interface association was done in an earlier step.

```
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg1
Router(config-l2vpn-bg)# bridge-domain bd1
Router(config-l2vpn-bg-bd)# interface Bundle-Ether101.101
Router(config-l2vpn-bg-bd-ac)# routed interface BVI101
```

Similar configurations are done for other interfaces.

```
Router(config-l2vpn-bg-bd)# interface Bundle-Ether102.101
Router(config-l2vpn-bg-bd-ac)# routed interface BVI101
.
Router(config-l2vpn-bg-bd)# interface HundredGigE0/11/0/25.101
Router(config-l2vpn-bg-bd-ac)# routed interface BVI101
.
Router(config-l2vpn-bg-bd)# interface HundredGigE0/11/0/31.101
Router(config-l2vpn-bg-bd-ac)# routed interface BVI101
Router(config-l2vpn-bg-bd-ac)# commit
```

MPLS Static Configuration

- BVI BVI101 is associated with the MPLS static LSP with a destination IP address 209.165.201.1.
- MPLS configurations (such as attaching an incoming label [1000101 and removing the MPLS label at the LSP egress device) are added.

```
Router# configure terminal
Router(config)# mpls static
Router(config-mpls-static)# lsp bvi-101
Router(config-mpls-static-lsp)# in-label 1000101 allocate
Router(config-mpls-static-lsp)# forward
Router(config-mpls-static-lsp-fwd)# path 1 resolve-nexthop 209.165.201.1 out-label pop
Router(config-mpls-static-lsp-fwd)# commit
```

Traffic Engineering

- Segment Routing (SR)is used for Traffic Engineering (TE).
- For the specified SR policy, the end point (or destination) is 209.165.201.1, same as the MPLS LSP destination for transporting L2VPN traffic.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# policy sr-static-mpls-to-LER2
Router(config-sr-te-policy)# binding-sid mpls 250100
Router(config-sr-te-policy)# color 100 end-point ipv4 209.165.201.1
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 100
Router(config-sr-te-policy-path-pref)# explicit segment-list to-LER2
Router(config-sr-te-policy-path-pref)# commit
```

A segment list is created with MPLS labels.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# segment-list to-LER2
Router(config-sr-te-sl)# index 10 mpls label 1000101
Router(config-sr-te-sl)# index 20 mpls label 300501
Router(config-sr-te-sl)# commit
```

Verification

Use this command to view MPLS static LSP details specific to the BVI:

```
Router# show mpls static lsp bvi-101 detail

LSP Name             Label   VRF             AFI  Type         Prefix           RW Configured
   Status
-------------------- ------- --------------- ---- ------------ ---------------- ------
bvi-101              1000101 default         N/A  X-Connect    N/A              Yes
      Created
  PRIMARY SET:
    [resolve-mode: nexthop 209.165.201.1]
    Path 0 : nexthop Bundle-Ether1 198.51.100.1, out-label Pop, Role: primary, Path-id: 2,
 Status: valid
    Path 1 : nexthop Bundle-Ether2 203.0.113.100, out-label Pop, Role: primary, Path-id:
1, Status: valid
```

Use this command to view MPLS static LSP details specific to the router:

```
Router# show mpls static local-label 1000101 detail

Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
```

```
1000101 default         X-Connect    N/A              Yes              Created

  PRIMARY SET:
    [resolve-mode: nexthop 209.165.201.1]
    Path 0 : nexthop Bundle-Ether1 198.51.100.1, out-label Pop, Role: primary, Path-id: 2,
 Status: valid
    Path 1 : nexthop Bundle-Ether2 203.0.113.100, out-label Pop, Role: primary, Path-id:
1, Status: valid
```

MPLS Over GRE Tunnels

Table 8: Feature History Table

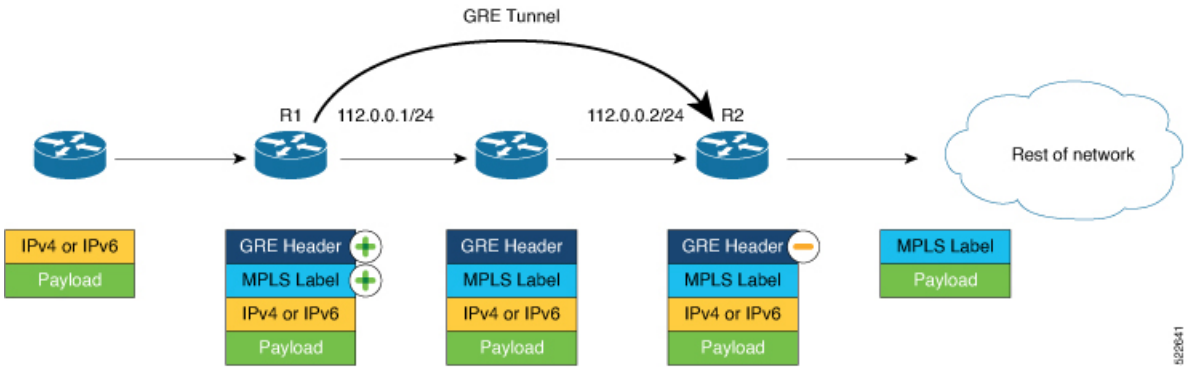
Feature Name	Release Information	Feature Description
MPLS Over GRE Tunnels	Release 7.5.2	The MPLS over generic routing encapsulation (MPLSoGRE) provides a mechanism for tunneling MPLS packets over a non-MPLS network. This feature uses MPLS over GRE to encapsulate MPLS packets inside GRE tunnels to create a virtual point-to-point link across non-MPLS networks. With this feature, the core network can be configured with IPv4 addresses to interconnect the MPLS networks through the IP network.

MPLS over GRE supports MPLS static forwarding over a GRE tunnel at line rate, which is the normal speed at which the traffic is sent through networks. For more information on line rate, see the Cisco 8000 Series Routers Data Sheet.

You can configure a provider router to send incoming customer traffic over the GRE tunnel, addressed to a set of load-balancing servers.

The GRE tunnel is configured with encapsulation that adds an MPLS packet and a GRE header to the incoming packet at the starting point of the tunnel. When the packet reaches the endpoint of the GRE tunnel, the GRE header is removed and the payload is forwarded to the destination based on the MPLS label.

Figure 12: MPLS Over GRE Tunnel



In the image, you can see that the GRE tunnel begins at router R1. R1 uses the policy based routing (PBR) process for GRE tunnel encapsulation, adds an MPLS label to the incoming packet, and then adds a GRE header. Then it sends the traffic towards router R2.

R2 uses the PBR process for GRE tunnel decapsulation, and based on the MPLS label, it forwards the traffic towards its destination.

Configuration Example

This example shows how to enable MPLS static forwarding over GRE tunnel.

Configuration on router R1.

1. Configure a tunnel interface.
2. Configure the mode of encapsulation as GRE for the tunnel interface.
3. Configure a policy map and redirect the traffic to next hop IP.
4. Assign the policy map to a VLAN subinterface.
5. Configure MPLS out labels to be applied to the incoming packets.

Configuration on router R2.

1. Configure a tunnel interface.
2. Configure the mode of decapsulation as GRE for the tunnel interface.
3. Configure a class map with match criteria for the source and destination IP addresses.
4. Configure a policy map and specify the class name configured in the class map.
5. Configure GRE decapsulation.

GRE Tunnel Configuration on R1

The GRE tunnel starts on R1.

The GRE tunnel destination must be a valid IPv4 address.

```
Router# configure
Router(config)# interface tunnel-ip1
Router(config-if)# ipv4 address 112.0.0.1 255.255.255.0
```

```
Router(config-if)# tunnel mode gre ipv4 encap
Router(config-if)# tunnel source Loopback 0
Router(config-if)# tunnel destination 50.0.0.1
Router(config-if)# commit
```

PBR Configuration for Encapsulation on R1

GRE encapsulation must be based on a policy map. Configure a policy map and redirect the traffic to next hop IP.

```
Router(config)# policy-map type pbr PBR_ENCAP_1
Router(config-pmap)# class type traffic class-default
Router(config-pmap-c)# redirect ipv4 nexthop 111.0.0.1
Router(config-pmap-c)# end-policy-map
```

Assign the policy map to one or more VLAN subinterfaces.

```
Router(config)# interface Bundle-Ether 111.1
Router(config-if)# service-policy type pbr input PBR_ENCAP_1
Router(config-if)# ipv4 address 50.0.2.1 255.255.255.0
Router(config-if)# encapsulation dot1q 1
```

MPLS Static Configuration on R1

Configure MPLS out labels. Ensure that the out label is the same for all paths of an MPLS static label switch path(LSP). You can configure up to a maximum of 16 paths. After applying the labels, the packets are forwarded to the specified next hop.

```
Router(config)# mpls static
Router(config-mpls-static)# lsp v4-encap-payload-1
Router(config-mpls-static-lsp)# in-label 10001 allocate per-prefix 111.0.0.1/32
Router(config-mpls-static-lsp)# forward
Router(config-mpls-static-lsp-fwd))# path 1 nexthop tunnel-ip1 out-label 20001
Router(config-mpls-static-lsp-fwd))# path 2 nexthop tunnel-ip2 out-label 20001
Router(config-mpls-static-lsp-fwd))# path 3 nexthop tunnel-ip3 out-label 20001
Router(config-mpls-static-lsp-fwd))# path 4 nexthop tunnel-ip4 out-label 20001
Router(config-mpls-static-lsp-fwd)# commit
```

GRE Tunnel Configuration on R2

The GRE tunnel stops on R2.

```
Router # configure
Router(config)# interface tunnel-ip1
Router(config-if)# ipv4 address 112.0.0.2 255.255.255.0
Router(config-if)# tunnel mode gre ipv4 decap
Router(config-if)# tunnel source Loopback 0
Router(config-if)# tunnel destination 10.0.0.1
Router(config-if)# commit
```

PBR Configuration for GRE Tunnel Decapsulation on R2

```
Router(config)# class-map type traffic match-all test_gre1
Router(config-cmap)# match protocol gre
Router(config-cmap)# match destination-address ipv4 50.0.0.1 255.255.255.255
Router(config-cmap)# match source-address ipv4 10.0.0.1 255.255.255.255
Router(config-cmap)# end-class-map
Router(config)# policy-map type pbr P1-test
```

```
Router(config-pmap)# class type traffic test_gre1
Router(config-pmap-c)#decapsulate gre
Router(config-pmap-c)# end-policy-map
Router(config)# vrf-policy vrf default address-family ipv4 policy type pbr input P1-test
```

Running Configuration

Use the following show commands to view the configuration.

Tunnel-IP configuration on R1

```
Router# show running-config interface tunnel-ip 1

interface tunnel-ip1
    ipv4 address 112.0.0.1 255.255.255.0
    tunnel mode gre ipv4 encap
    tunnel source Loopback 0
    tunnel destination 50.0.0.1
!
```

PBR Configuration for GRE Tunnel Encapsulation on R1

```
Router# show running-config policy-map type pbr *

policy-map type pbr PBR_ENCAP_1
 class type traffic class-default
  redirect ipv4 nexthop 111.0.0.1
 !
 end-policy-map
 !
```

MPLS Static Configuration on R1

```
Router# show running-config mpls static

mpls static
  lsp v4-encap-payload-1
     in-label 10001 allocate per-prefix 111.0.0.1/32
     forward
        path 1 nexthop tunnel-ip1 out-label 20001
        path 2 nexthop tunnel-ip2 out-label 20001
        path 3 nexthop tunnel-ip3 out-label 20001
        path 4 nexthop tunnel-ip4 out-label 20001
        !
```

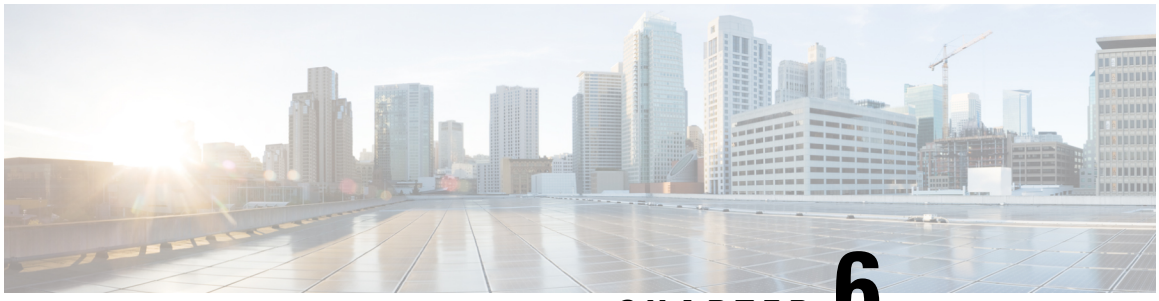
Tunnel-IP Configuration on R2

```
Router# show running-config int tunnel-ip 1

interface tunnel-ip1
 ipv4 address 112.0.0.2 255.255.255.0
 tunnel mode gre ipv4 decap
 tunnel source Loopback 0
 tunnel destination 10.0.0.1
!
```

PBR Configuration for GRE Tunnel Decapsulation on R2

```
Router# show running-config class-map type traffic match-all

class-map type traffic match-all test_gre1
```

```
    match protocol gre
    match destination-address ipv4 50.0.0.1 255.255.255.255
    match source-address ipv4 10.0.0.1 255.255.255.255
end-class-map
!
policy-map type pbr P1-test
   class type traffic test_gre1
   decapsulate gre
!
   class type traffic class-default
!
end-policy-map
!

vrf-policy
vrf default address-family ipv4 policy type pbr input P1-test
!
```



CHAPTER 6

Implementing MPLS Traffic Engineering

Traditional IP routing emphasizes on forwarding traffic to the destination as fast as possible. As a result, the routing protocols find out the least-cost route according to its metric to each destination in the network and every router forwards the packet based on the destination IP address and packets are forwarded hop-by-hop. Thus, traditional IP routing does not consider the available bandwidth of the link. This can cause some links to be over-utilized compared to others and bandwidth is not efficiently utilized. Traffic Engineering (TE) is used when the problems result from inefficient mapping of traffic streams onto the network resources. Traffic engineering allows you to control the path that data packets follow and moves traffic flows from congested links to non-congested links that would not be possible by the automatically computed destination-based shortest path.

Multiprotocol Label Switching (MPLS) with its label switching capabilities, eliminates the need for an IP route look-up and creates a virtual circuit (VC) switching function, allowing enterprises the same performance on their IP-based network services as with those delivered over traditional networks such as Frame Relay or Asynchronous Transfer Mode (ATM). MPLS traffic engineering (MPLS-TE) relies on the MPLS backbone to replicate and expand upon the TE capabilities of Layer 2 ATM and Frame Relay networks.

MPLS-TE learns the topology and resources available in a network and then maps traffic flows to particular paths based on resource requirements and network resources such as bandwidth. MPLS-TE builds a unidirectional tunnel from a source to a destination in the form of a label switched path (LSP), which is then used to forward traffic. The point where the tunnel begins is called the tunnel headend or tunnel source, and the node where the tunnel ends is called the tunnel tailend or tunnel destination. A router through which the tunnel passes is called the mid-point of the tunnel.

MPLS uses extensions to a link-state based Interior Gateway Protocol (IGP), such as Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF). MPLS calculates TE tunnels at the LSP head based on required and available resources (constraint-based routing). If configured, the IGP automatically routes the traffic onto these LSPs. Typically, a packet that crosses the MPLS-TE backbone travels on a single LSP that connects the ingress point to the egress point. MPLS TE automatically establishes and maintains the LSPs across the MPLS network by using the Resource Reservation Protocol (RSVP).

Note Combination of unlabelled paths protected by labelled paths is not supported.

Prerequisites for Implementing Cisco MPLS Traffic Engineering

These prerequisites are required to implement MPLS TE:

- Router that runs Cisco IOS XR software .
- Installed composite mini-image and the MPLS package, or a full composite image.
- IGP activated.

Overview of MPLS-TE Features

In MPLS traffic engineering, IGP extensions flood the TE information across the network. Once the IGP distributes the link attributes and bandwidth information, the headend router calculates the best path from head to tail for the MPLS-TE tunnel. This path can also be configured explicitly. Once the path is calculated, RSVP-TE is used to set up the TE LSP (Labeled Switch Path).

To forward the traffic, you can configure autoroute, forward adjacency, or static routing. The autoroute feature announces the routes assigned by the tailend router and its downstream routes to the routing table of the headend router and the tunnel is considered as a directly connected link to the tunnel.

If forward adjacency is enabled, MPLS-TE tunnel is advertised as a link in an IGP network with the link's cost associated with it. Routers outside of the TE domain can see the TE tunnel and use it to compute the shortest path for routing traffic throughout the network.

MPLS-TE provides protection mechanism known as fast reroute to minimize packet loss during a failure. For fast reroute, you need to create back up tunnels. The autotunnel backup feature enables a router to dynamically build backup tunnels when they are needed instead of pre-configuring each backup tunnel and then assign the backup tunnel to the protected interfaces.

DiffServ Aware Traffic Engineering (DS-TE) enables you to configure multiple bandwidth constraints on an MPLS-enabled interface to support various classes of service (CoS). These bandwidth constraints can be treated differently based on the requirement for the traffic class using that constraint.

The MPLS traffic engineering auto-tunnel mesh feature allows you to set up full mesh of TE tunnels automatically with a minimal set of MPLS traffic engineering configurations. The MPLS-TE auto bandwidth feature allows you to automatically adjusts bandwidth based on traffic patterns without traffic disruption.

The MPLS-TE interarea tunneling feature allows you to establish TE tunnels spanning multiple Interior Gateway Protocol (IGP) areas and levels, thus eliminating the requirement that headend and tailend routers should reside in a single area.

For detailed information about MPLS-TE features, see the *MPLS-TE Features - Details* topic.

Note MPLS-TE Nonstop Routing (NSR) is enabled by default without any user configuration and cannot be disabled.

MPLS-TE NSR means the application is in hot-standby mode and standby MPLS-TE instance is ready to take over from the active instance quickly on RP failover.

Note that the MPLS-TE does not do routing. If there is standby card available then the MPLS-TE instance is in a hot-standby position.

The following output shows the status of MPLS-TE NSR:

```
Router#show mpls traffic-eng nsr status

TE Process Role        : V1 Active
Current Status      : Ready
    Ready since        : Tue Nov 01 10:42:34 UTC 2022 (1w3d ago)
   IDT started         : Tue Nov 01 03:28:48 UTC 2022 (1w3d ago)
    IDT ended          : Tue Nov 01 03:28:48 UTC 2022 (1w3d ago)
Previous Status        : Not ready
    Not ready reason   : Collaborator disconnected
    Not ready since    : Tue Nov 01 10:42:34 UTC 2022 (1w3d ago)
```

During any issues with the MPLS-TE, the NSR on the router gets affected which is displayed in the show redundancy output as follows:

```
Router#show mpls traffic-eng nsr status details
.
.
.

Current active rmf state: 4 (I_READY)
All standby not-ready bits clear - standby should be ready

Current active rmf state for NSR: Not ready
<jid> <node> <name> Reason for standby not NSR-ready
1082 0/RP0/CPU0 te_control TE NSR session not synchronized
Not ready set Wed Nov 19 17:28:14 2022: 5 hours, 23 minutes ago
1082 0/RP1/CPU0 te_control Standby not connected
Not ready set Wed Nov 19 17:29:11 2022: 5 hours, 22 minutes ago
```

How MPLS-TE Works

MPLS-TE automatically establishes and maintains label switched paths (LSPs) across the backbone by using RSVP. The path that an LSP uses is determined by the LSP resource requirements and network resources, such as bandwidth. Available resources are flooded by extensions to a link state based Interior Gateway Protocol (IGP). MPLS-TE tunnels are calculated at the LSP headend router, based on a fit between the required and available resources (constraint-based routing). The IGP automatically routes the traffic to these LSPs. Typically, a packet crossing the MPLS-TE backbone travels on a single LSP that connects the ingress point to the egress point.

The following sections describe the components of MPLS-TE:

Tunnel Interfaces

From a Layer 2 standpoint, an MPLS tunnel interface represents the headend of an LSP. It is configured with a set of resource requirements, such as bandwidth and media requirements, and priority. From a Layer 3 standpoint, an LSP tunnel interface is the headend of a unidirectional virtual link to the tunnel destination.

MPLS-TE Path Calculation Module

This calculation module operates at the LSP headend. The module determines a path to use for an LSP. The path calculation uses a link-state database containing flooded topology and resource information.

RSVP with TE Extensions

RSVP operates at each LSP hop and is used to signal and maintain LSPs based on the calculated path.

MPLS-TE Link Management Module

This module operates at each LSP hop, performs link call admission on the RSVP signaling messages, and keep track on topology and resource information to be flooded.

Link-state IGP

Either Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF) can be used as IGPs. These IGPs are used to globally flood topology and resource information from the link management module.

Label Switching Forwarding

This forwarding mechanism provides routers with a Layer 2-like ability to direct traffic across multiple hops of the LSP established by RSVP signaling.

Soft-Preemption

MPLS-TE preemption consists of freeing the resources of an established LSP, and assigning them to a new LSP. The freeing of resources causes a traffic disruption to the LSP that is being preempted. Soft preemption is an extension to the RSVP-TE protocol to minimize and even eliminate such traffic disruption over the preempted LSP.

The soft-preemption feature attempts to preempt the LSPs in a graceful manner to minimize or eliminate traffic loss. However, the link might be over-subscribed for a period of time.

In a network that implements soft preemption, zero traffic loss is achieved in this manner:

- When signaling a new LSP, the ingress router indicates to all the intermediate nodes that the existing LSP is to be softly preempted, in case its resources are needed and is to be reassigned.
- When a given intermediate node needs to soft-preempt the existing LSP, it sends a new or special path error (preemption pending) to the ingress router. The intermediate node does not dismantle the LSP and maintains its state.
- When the ingress router receives the path error (preemption pending) from the intermediate node, it immediately starts a re-optimization that avoids the link that caused the preemption.
- When the re-optimization is complete, the ingress router tears down the soft-preempted LSP.

Soft-preemption over FRR Backup Tunnels

The soft-preemption over FRR backup tunnels feature enables to move LSP traffic over the backup tunnels when the LSP is soft- preempted. MPLS TE tunnel soft-preemption allows removal of extra TE traffic in a graceful manner, by giving the preempted LSP a grace period to move away from the link. Though this mechanism saves the traffic of the preempted LSP from being dropped, this might cause traffic drops due to congestion as more bandwidth is reserved on the link than what is available. When the soft-preemption over FRR backup tunnel is enabled, the traffic of the preempted LSP is moved onto the FRR backup, if it is available and ready. This way, the capacity of the backup tunnel is used to remove the potential congestion that might be caused by soft-preemption.

SRLG Limitations

There are few limitations to the configured SRLG feature:

- The **exclude-address** and **exclude-srlg** options are not allowed in the IP **explicit path strict-address** network.
- Whenever SRLG values are modified after tunnels are signaled, they are verified dynamically in the next path verification cycle.

RSVP-TE Dark Bandwidth Accounting

This section describes the RSVP-TE Dark Bandwidth Accounting feature that allows for the co-existence of non-zero bandwidth RSVP-TE tunnels and Segment Routing (SR) in the same network domain. This feature measures dark bandwidth traffic and accounts for it in the RSVP-TE bandwidth reservations to avoid overbooking the links in the network.

Dark bandwidth is the actual utilization of the link by the subset of the traffic that is not explicitly admission controlled by RSVP-TE. Dark bandwidth is not considered during path computation and admission control for distributed RSVP-TE LSPs.

In this solution, SR is assumed to be the main source of dark bandwidth on the links in the network. In addition, SR traffic is considered to have a higher priority than any other traffic transported by RSVP-TE LSPs. Therefore, the bandwidth consumed by SR effectively reduces the link bandwidth available to RSVP-TE LSPs.

The RSVP-TE Dark Bandwidth Accounting feature consists of the following:

- The measurement of SR traffic on interfaces via new per-interface aggregate SR counters
- The calculation of dark bandwidth rate based on the measured SR traffic statistics
- The calculation of the RSVP-TE effective maximum reservable bandwidth (BMRe).

  The BMRe is used for the purpose of pre-emption as well as advertisement (flooding) via IGP. A threshold is evaluated before triggering flooding.

Computing the Dark Bandwidth and RSVP-TE Effective Maximum Reservable Bandwidth

The statistics collector process (statsD) is responsible for returning statistics counters for each feature. For each traffic engineering (TE)-enabled interface, the TE process collects new SR bandwidth rate statistics (samples) from the statsD process, within a specified sampling interval. These samples are collected over a period of time called an application interval.

After each application interval, the average value of the collected rate samples is used to compute the dark bandwidth rate and the BMRe rate.

The following example shows how the BMRe is computed (assuming a link capacity of 10Gbps and a configured BMR [BMRc] of 90%):

- Link capacity = 10Gbps
- BMRc = RSVP percentage of link capacity = 9Gbps
- Calculated dark bandwidth rate = 2Gpbs
- BMRe = 7Gbps

In this example, the bandwidth available for RSVP-TE LSP admission is 7Gbps. This value is flooded in the network if the flooding threshold is crossed.

Note When you change the RSVP bandwidth percentage configuration or when the bundle capacity changes due to bundle-member state change, TE accounts for the dark bandwidth when new bandwidth values are advertised.

Note The measured dark bandwidth can be increased or decreased based on a configurable adjustment factor.

When the dark bandwidth rate increases for a link, it will lower the BMR of that link, which might trigger preemption of the RSVP-TE LSPs. Preemption consists of freeing the resources of an established LSP, and assigning them to a new LSP. The freeing of resources causes a traffic disruption to the LSP that is being preempted.

Point-to-Multipoint Traffic-Engineering

This section contains the following topics:

Point-to-Multipoint Traffic-Engineering Overview

The Point-to-Multipoint (P2MP) Resource Reservation Protocol-Traffic Engineering (RSVP-TE) solution allows service providers to implement IP multicast applications, such as IPTV and real-time video, broadcast over the MPLS label switch network. The RSVP-TE protocol is extended to signal point-to-point (P2P) and P2MP label switched paths (LSPs) across the MPLS network.

Note

- For P2MP tunnels, a Cisco 8000 Series router supports the mid-point router function, and does not support source or receiver functions. To know how to configure a source or receiver (destination) router in a P2MP tunnel, refer the MPLS configuration guide for the corresponding platform.
- The FRR function is not supported for P2MP tunnels.

By using RSVP-TE extensions as defined in RFC 4875, multiple subLSPs are signaled for a given TE source. The P2MP tunnel is considered as a set of Source-to-Leaf (S2L) subLSPs that connect the TE source to multiple leaf Provider Edge (PE) nodes.

At the TE source, the ingress point of the P2MP-TE tunnel, IP multicast traffic is encapsulated with a unique MPLS label, which is associated with the P2MP-TE tunnel. The traffic continues to be label-switched in the P2MP tree. If needed, the labeled packet is replicated at branch nodes along the P2MP tree. When the labeled packet reaches the egress leaf (PE) node, the MPLS label is removed and forwarded onto the IP multicast tree across the PE-CE link.

To enable end-to-end IP multicast connectivity, RSVP is used in the MPLS-core for P2MP-TE signaling and PIM is used for PE-CE link signaling.

- All edge routers are running PIM-SSM or Source-Specific Multicast (SSM) to exchange multicast routing information with the directly-connected Customer Edge (CE) routers.
- In the MPLS network, RSVP P2MP-TE replaces PIM as the tree building mechanism, RSVP-TE grafts or prunes a given P2MP tree when the end-points are added or removed in the TE source configuration (explicit user operation).

These are the definitions for Point-to-Multipoint (P2MP) tunnels. Cisco 8000 Series routers only support the role of a mid-point.

Source

Configures the node in which Label Switched Path (LSP) signaling is initiated.

Mid-point

Specifies the transit node in which LSP signaling is processed (for example, not a source or receiver).

Receiver, Leaf, and Destination

Specifies the node in which LSP signaling ends.

Branch Point

Specifies the node in which packet replication is performed.

Source-to-Leaf (S2L) SubLSP

Specifies the P2MP-TE LSP segment that runs from the source to one leaf.

Point-to-Multipoint Traffic-Engineering Features

- P2MP RSVP-TE (RFC 4875) is supported. RFC 4875 is based on nonaggregate signaling; for example, per S2L signaling. Only P2MP LSP is supported.
- The **interface tunnel-mte** command identifies the P2MP interface type.
- P2MP tunnel setup is supported with label replication.

- Explicit routing is supported by using under utilized links.
- Reoptimization is supported by calculating a better set of paths to the destination with no traffic loss.

Note Per-S2L reoptimization is not supported.

- IPv4 and IPv6 payloads are supported.
- IPv4 and IPv6 multicast forwarding are supported on a P2MP tunnel interface through a static IGMP and MLD group configuration.
- Both IP multicast and P2MP Label Switch Multicast (LSM) coexist in the same network; therefore, both use the same forwarding plane (LFIB or MPLS Forwarding Infrastructure [MFI]).
- P2MP label replication supports only Source-Specific Multicast (SSM) traffic. SSM configuration supports the default value, none.
- Static mapping for multicast groups to the P2MP-TE tunnel is required.

Point-to-Multipoint Traffic-Engineering Benefits

- Single point of traffic control ensures that signaling and path engineering parameters (for example, protection and diversity) are configured only at the TE source node.
- Ability to configure explicit paths to enable optimized traffic distribution and prevention of single point of failures in the network.
- Link protection of MPLS-labeled traffic traversing branch paths of the P2MP-TE tree.
- Ability to do bandwidth Admission Control (AC) during set up and signaling of P2MP-TE paths in the MPLS network.

Point-to-Multipoint RSVP-TE

RSVP-TE signals a P2MP tunnel base that is based on a manual configuration. If all Source-to-Leaf (S2L)s use an explicit path, the P2MP tunnel creates a static tree that follows a predefined path based on a constraint such as a deterministic Label Switched Path (LSP). If the S2L uses a dynamic path, RSVP-TE creates a P2MP tunnel base on the best path in the RSVP-TE topology. RSVP-TE supports bandwidth reservation for constraint-based routing.

RSVP-TE distributes stream information in which the topology tree does not change often (where the source and receivers are). For example, large scale video distribution between major sites is suitable for a subset of multicast applications. Because multicast traffic is already in the tunnel, the RSVP-TE tree is protected as long as you build a backup path.

The P2MP tunnel is signaled by the dynamic and explicit path option in the IGP intra area. Only interArea and interAS, which are used for the P2MP tunnels, are signaled by the verbatim path option.

Point-to-Multipoint Label Switch Path

The Point-to-Multipoint Label Switch Path (P2MP LSP) has only a single root, which is the Ingress Label Switch Router (LSR). The P2MP LSP is created based on a receiver that is connected to the Egress LSR. The Egress LSR initiates the creation of the tree (for example, tunnel grafting or pruning is done by performing an individual sub-LSP operation) by creating the Forwarding Equivalency Class (FEC) and Opaque Value.

Note Grafting and pruning operate on a per destination basis.

The Opaque Value contains the stream information that uniquely identifies the tree to the root. To receive label switched multicast packets, the Egress Provider Edge (PE) indicates to the upstream router (the next hop closest to the root) which label it uses for the multicast source by applying the label mapping message.

The upstream router does not need to have any knowledge of the source; it needs only the received FEC to identify the correct P2MP LSP. If the upstream router does not have any FEC state, it creates it and installs the assigned downstream outgoing label into the label forwarding table. If the upstream router is not the root of the tree, it must forward the label mapping message to the next hop upstream. This process is repeated hop-by-hop until the root is reached.

By using downstream allocation, the router that wants to receive the multicast traffic assigns the label for it. The label request, which is sent to the upstream router, is similar to an unsolicited label mapping (that is, the upstream does not request it). The upstream router that receives that label mapping uses the specific label to send multicast packets downstream to the receiver. The advantage is that the router, which allocates the labels, does not get into a situation where it has the same label for two different multicast sources. This is because it manages its own label space allocation locally.

Path Option for Point-to-Multipoint RSVP-TE

P2MP tunnels are signaled by using the dynamic and explicit path-options in an IGP intra area. InterArea cases for P2MP tunnels are signaled by the verbatim path option.

Path options for P2MP tunnels are individually configured for each sub-LSP. Only one path option per sub-LSP (destination) is allowed. You can choose whether the corresponding sub-LSP is dynamically or explicitly routed. For the explicit option, you can configure the verbatim path option to bypass the topology database lookup and verification for the specified destination.

Both dynamic and explicit path options are supported on a per destination basis by using the **path-option (P2MP-TE)** command. In addition, you can combine both path options.

Explicit Path Option

Configures the intermediate hops that are traversed by a sub-LSP going from the TE source to the egress MPLS node. Although an explicit path configuration enables granular control sub-LSP paths in an MPLS network, multiple explicit paths are configured for specific network topologies with a limited number of (equal cost) links or paths.

Dynamic Path Option

Computes the IGP path of a P2MP tree sub-LSP that is based on the OSPF and ISIS algorithm. The TE source is dynamically calculated based on the IGP topology.

Note Dynamic path option can only compute fully-diverse standby paths. While, explicit path option supports partially diverse standby paths as well.

Dynamic Path Calculation Requirements

Dynamic path calculation for each sub-LSP uses the same path parameters as those for the path calculation of regular point-to-point TE tunnels. As part of the sub-LSP path calculation, the link resource (bandwidth) is included, which is flooded throughout the MPLS network through the existing RSVP-TE extensions to OSPF and ISIS. Instead of dynamic calculated paths, explicit paths are also configured for one or more sub-LSPs that are associated with the P2MP-TE tunnel.

- OSPF or ISIS are used for each destination.
- TE topology and tunnel constraints are used to input the path calculation.
- Tunnel constraints such as affinity, bandwidth, and priorities are used for all destinations in a tunnel.
- Path calculation yields an explicit route to each destination.

Static Path Calculation Requirements

The static path calculation does not require any new extensions to IGP to advertise link availability.

- Explicit path is required for every destination.
- Offline path calculation is used.
- TE topology database is not needed.
- If the topology changes, reoptimization is not required.

Point-to-Multipoint Implicit Null

The Point-to-Multipoint (P2MP) implicit null feature enables the forwarding of unicast traffic over P2MP tunnels. This feature is enabled by default and requires no configuration.

In a P2MP tunnel, the tailend router signals the implicit null label to the midpoint router. If the given MPI leg of the P2MP tunnel is implicit null capable (where the penultimate router is capable to do penultimate hop popping), the FIB (Forwarding Information Base) creates two NRLDI (Non Recursive Load Distribution Index) entries, one for forwarding the IPv6 labeled packets, and the other for non-labeled IPv4 unicast traffic.

The headend and the tailend routers handle the unicast traffic arriving on the P2MP tunnel. The midpoint router forwards the unicast traffic to its bud and tailend routers.

The use of implicit null at the end of a tunnel is called penultimate hop popping (PHP). The FIB entry for the tunnel on the PHP router shows a "pop label" as the outgoing label.

In some cases, it could be that the packets have two or three or more labels in the label stack. Then the implicit null label used at the tailend router would signal the penultimate hop router to pop one label and send the labeled packet with one label less to the tailend router. Then the tailend router does not have to perform two label lookups. The use of the implicit null label does not mean that all labels of the label stack must be removed; only one label is "popped" off (remove the top label on the stack). In any case, the use of the implicit null label prevents the tailend router from performing two lookups.

Restriction - The P2MP implicit null feature may cause multicast traffic drop with implicit null label on the tailend routers. This is because the P2MP implicit null feature does not support forwarding of multicast traffic when no label is received on the tailend router.

Configuring MPLS-TE

MPLS-TE requires co-ordination among several global neighbor routers. RSVP, MPLS-TE and IGP are configured on all routers and interfaces in the MPLS traffic engineering network. Explicit path and TE tunnel interfaces are configured only on the head-end routers. MPLS-TE requires some basic configuration tasks explained in this section.

Building MPLS-TE Topology

Building MPLS-TE topology, sets up the environment for creating MPLS-TE tunnels. This procedure includes the basic node and interface configuration for enabling MPLS-TE. To perform constraint-based routing, you need to enable OSPF or IS-IS as IGP extension.

Before You Begin

Before you start to build the MPLS-TE topology, the following pre-requisites are required:

- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.
- Enable RSVP on the port interface.

Example

This example enables MPLS-TE on a node and then specifies the interface that is part of the MPLS-TE. Here, OSPF is used as the IGP extension protocol for information distribution.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface hundredGigE 0/0/0/3
Router(config)# router ospf area 1
Router(config-ospf)# area 0
Router(config-ospf-ar)# mpls traffic-eng
Router(config-ospf-ar)# interface hundredGigE 0/0/0/3
Router(config-ospf-ar-if)# exit
Router(config-ospf)# mpls traffic-eng router-id 192.168.70.1
Router(config)# commit
```

Example

This example enables MPLS-TE on a node and then specifies the interface that is part of the MPLS-TE. Here, IS-IS is used as the IGP extension protocol for information distribution.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface hundredGigE 0/0/0/3
Router(config)# router isis 1
Router(config-isis)# net 47.0001.0000.0000.0002.00
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
```

```
Router(config-isis-af)# mpls traffic-eng level 1
Router(config-isis-af)# exit
Router(config-isis)# interface hundredGigE 0/0/0/3
Router(config-isis-if)# exit
Router(config)# commit
```

Configuring Automatic Bandwidth

Automatic bandwidth allows you to dynamically adjust bandwidth reservation based on measured traffic. MPLS-TE automatic bandwidth monitors the traffic rate on a tunnel interface and resizes the bandwidth on the tunnel interface to align it closely with the traffic in the tunnel. MPLS-TE automatic bandwidth is configured on individual Label Switched Paths (LSPs) at every headend router.

Adjustment Threshold - It is defined as a percentage of the current tunnel bandwidth and an absolute (minimum) bandwidth. Both thresholds must be fulfilled for the automatic bandwidth to resignal the tunnel. The tunnel bandwidth is resized only if the difference between the largest sample output rate and the current tunnel bandwidth is larger than the adjustment thresholds.

For example, assume that the automatic bandwidth is enabled on a tunnel in which the highest observed bandwidth B is 30 Mbps. Also, assume that the tunnel was initially configured for 45 Mbps. Therefore, the difference is 15 mbit/s. Now, assuming the default adjustment thresholds of 10% and 10kbps, the tunnel is signaled with 30 Mbps when the application timer expires. This is because 10% of 45Mbit/s is 4.5 Mbit/s, which is smaller than 15 Mbit/s. The absolute threshold, which by default is 10kbps, is also crossed.

The following table specifies the parameters that can be configured as part of automatic bandwidth configuration.

Table 9: Automatic Bandwidth Parameters

Bandwidth Parameters	Description
Application frequency	Configures how often the tunnel bandwidths changed for each tunnel. The default value is 24 hours.
Bandwidth limit	Configures the minimum and maximum automatic bandwidth to set on a tunnel.
Bandwidth collection frequency	Enables bandwidth collection without adjusting the automatic bandwidth. The default value is 5 minutes.
Overflow threshold	Configures tunnel overflow detection.
Adjustment threshold	Configures the tunnel-bandwidth change threshold to trigger an adjustment.

Adjustment Threshold

Configuration Example

This example enables automatic bandwidth on MPLS-TE tunnel interface and configure the following automatic bandwidth variables.

- Application frequency
- Bandwidth limit

- Adjustment threshold
- Overflow detection

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# auto-bw
Router(config-if-tunte-autobw)# application 1000
Router(config-if-tunte-autobw)# bw-limit min 30 max 1000
Router(config-if-tunte-autobw)# adjustment-threshold 50 min 800
Router(config-if-tunte-autobw)# overflow threshold 100 limit 1
Router(config)# commit
```

Verification

Verify the automatic bandwidth configuration using the **show mpls traffic-eng tunnels auto-bw brief** command.

```
Router# show mpls traffic-eng tunnels auto-bw brief

  Tunnel    LSP  Last appl  Requested  Signalled    Highest    Application
  Name       ID   BW(kbps)   BW(kbps)   BW(kbps)   BW(kbps)      Time Left
  -------------- ------ ---------- ---------- ---------- ---------- --------------
  tunnel-te1      5          500         300        420          1h 10m
```

Configuring Automatic Capacity With Load-Interval Configuration

Table 10: Feature History Table

Feature Name	Release Information	Feature Description
Configuring Automatic Capacity With Load-Interval Configuration	Release 7.3.3	With this feature, you can enable the load-interval configuration for a main tunnel's clones, along with the automatic capacity feature.

The auto-bandwidth feature resizes MPLS-TE tunnels based on traffic loads. Multiple auto-bandwidth tunnels can be created for balancing traffic loads and redundancy.

The auto-capacity feature is an extension of the auto-bandwidth feature. With auto-capacity, for an auto-bandwidth enabled MPLS-TE tunnel, you can enable automatic creation and deletion of tunnels based on real-time capacity demands. These tunnels are called *clones*. For a main TE tunnel, you can specify the minimum and maximum number of clones, and allocate a nominal tunnel bandwidth value. Clones are automatically added to, or removed from, the main TE tunnel, based on the nominal bandwidth.

Consider the auto-capacity configuration example:

```
Router(config)# mpls traffic-eng
Router(config-mpls-te)# named-tunnels tunnel-te YOW2YZZ


Router(config-te-tun-name)# load-interval 90
Router(config-te-tun-name)# auto-bw
Router(config-mpls-te-tun-autobw)# auto-capacity
Router(config-mpls-te-tun-autobw)# commit
```

retries to establish the LSPs. The timeout range is 1 to 600 seconds.

The auto-capacity function is disabled by default. Since auto-bandwidth and auto-capacity functions are inter-related, these are the corresponding changes in behavior:

- When you enable auto-capacity, it is associated with a specific TE-tunnel, under the auto-bandwidth function. When you disable auto-bandwidth, auto-capacity is also disabled.
- If the load interval is enabled for the main tunnel, it is automatically applied to its clones too. For a main tunnel, if the auto-capacity feature is enabled but a load interval is not enabled, the clones' load interval value is set to a default of 300 seconds.

Splitting and Merging Tunnels

When there is a change in demand for bandwidth, MPLS-TE adds or reduces the number of tunnels and resizes the bandwidth of all the tunnels. It verifies these rules during this activity.

1. The number of tunnels between the headend-tailend router pair is within the specified range, and the bandwidth per tunnel is within the auto-bandwidth range.
2. The (*Bandwidth-per-tunnel*)*(*Number-of-Tunnels*) >= *Total-tunnel-bandwidth* requirement.

   While Rule 1 is enforced, MPLS-TE attempts to enforce Rule 2.
3. When the split requirement is met, and the maximum number of clones is not reached, at least one extra clone is added.

   When the merge requirement is met, and the minimum number of clones is not reached, at least one clone is removed
4. The nominal bandwidth value is used to balance the requirements of: (a) Number of tunnels and (b) Bandwidth for each tunnel. This helps in avoiding a merging or splitting instance at the next application event.

Configurations

```
/* Automatic Capacity Function */

Router# configure
Router(config)# mpls traffic-eng
```

The auto-capacity feature is only valid for the main tunnel **YOW2YZZ**, with reference to which, clones are created or removed.

```
Router(config-mpls-te)# named-tunnels tunnel-te YOW2YZZ
Router(config-te-tun-name)# auto-bw auto-capacity
```

MPLS-TE maintains the number of clones between 1 and 7. Including the main tunnel **YOW2YZZ**, the tunnel count range is between 2 and 8.

```
Router(config-te-tun-autocapacity)# max-clones 7
Router(config-te-tun-autocapacity)# min-clones 1
```

The nominal-bandwidth option is used for specifying the target bandwidth based on which MPLS-TE calculates the number of required tunnels.

```
Router(config-te-tun-autocapacity)# nominal-bandwidth 2000000
```

MPLS-TE also uses the merge-bandwidth and split-bandwidth values when implementing the auto-capacity feature.

```
Router(config-te-tun-autocapacity)# merge-bandwidth 1000000
Router(config-te-tun-autocapacity)# split-bandwidth 3000000
Router(config-te-tun-autocapacity)# commit
```

Verification

```
/* View the Auto-Capacity Feature Configuration */

Router# show mpls traffic-eng tunnels name YOW2YZZ

Name: YOW2YZZ     Ifhandle:0xf000014
  ..
  Config Parameters:
  ..
    Load-interval: 300 seconds  ..
Auto-Capacity: Enabled
  Minimum Clones: 1; Maximium Clones: 7
  Nominal BW: 2000000 kbps; Merge BW: 1000000 kbps; Split BW: 3000000 kbps
  Statistics:
    Splits: 0; Merges: 0
    Clones Created: 1; Clones Deleted: 0
    Clones High Watermark: 1
  Number of clones: 1
    Clone: YOW2YZZ-1
      Created: Thu Jan 27 13:55:06 2022; State: down
  ..
```

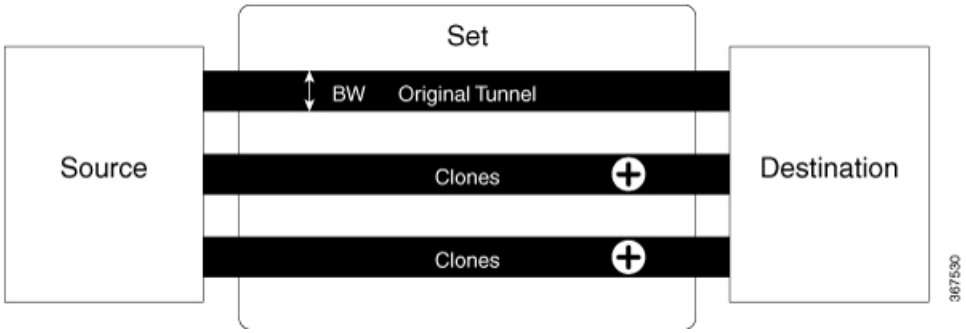
Configuring Auto-Bandwidth Bundle TE++

An MPLS-TE tunnel sets up labeled connectivity and provides dynamic bandwidth capacity between its endpoints. The auto-bandwidth function addresses the dynamic bandwidth capacity demands by resizing the MPLS-TE tunnels based on the measured traffic loads. However, many customers require multiple auto-bandwidth tunnels between two endpoints for load balancing and redundancy. The auto-bandwidth bundle TE++ function is an extension of the auto-bandwidth feature, and provides this support. When the aggregate bandwidth between the endpoints changes, MPLS-TE creates new tunnels or removes existing tunnels to load balance the traffic.

When MPLS-TE automatically creates new tunnels to meet increasing bandwidth demands, they are called clones. The original tunnel and its clones collectively form a *set*. The clones inherit the properties of the main tunnel, except for the user-configured load-interval value. You can specify an upper limit and lower limit on the number of clones.

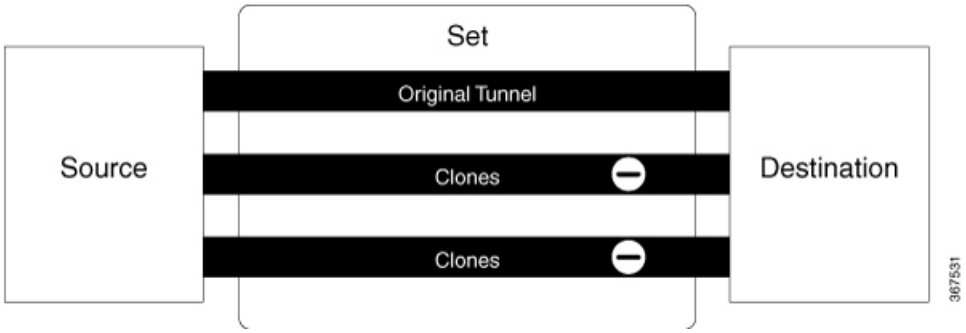
Splitting is the process of creating a new clone. When the bandwidth of a tunnel crosses the split bandwidth value, MPLS-TE creates a clone.

The following figure explains how MPLS-TE creates clones when the split bandwidth exceeds the configured value.



Merging is the process of removing a clone. If the bandwidth goes below the merge bandwidth value in a set of tunnels, MPLS-TE removes a clone.

The following figure explains how MPLS-TE removes clones when the bandwidth falls below the merge bandwidth value.



There are multiple ways to *load-share* the aggregate bandwidth demand among the tunnels in a set. An algorithm chooses the pair that satisfies the aggregate bandwidth requirements. You can configure a nominal bandwidth to guide the algorithm that determines the average bandwidth of the tunnels. If you don't configure, MPLS-TE uses the average of the split bandwidth and merge bandwidth values as the nominal bandwidth.

Restrictions and Guidelines

The following guidelines and restrictions apply for the auto-bandwidth bundle TE++ feature.

- This feature only supports named tunnels, and doesn't support tunnel-te interfaces.
- The range for the lower limit on the number of clones is 0–63. The default value is 0. The upper limit range is 1–63. The default value is 63.

Configure Auto-Bandwidth Bundle TE++

Configure the following parameters:

- **min-clones**: Specifies the minimum number of clones that the original tunnel can create.
- **max-clones**: Specifies the maximum number of clones that the original tunnel can create.
- **nominal-bandwidth**: Specifies the average bandwidth for computing the number of tunnels to satisfy the overall demand.

- **split-bandwidth**: Specifies the bandwidth for splitting the original tunnel. If the tunnel bandwidth exceeds the configured split bandwidth, MPLS-TE creates clones.
- **merge-bandwidth**: Specifies the bandwidth for merging clones with the original tunnel. If the bandwidth goes below the merge bandwidth value, MPLS-TE removes the clones.

Configuration Example: Named MPLS-TE Tunnel

This example shows how to configure the auto-bandwidth bundle TE++ feature for a named MPLS-TE tunnel.

Here, the lower and upper limits on the number of clones are two and four, respectively. The bandwidth size for splitting and merging are 200 kbps and 100 kbps, respectively.

```
Router(config)# mpls traffic-eng
Router(config-mpls-te)# named-tunnels
Router(config-te-named-tunnels)# tunnel-te xyz
Router(config-te-tun-name)# auto-bw
Router(config-mpls-te-tun-autobw)# auto-capacity
Router(config-te-tun-autocapacity)# min-clones 2
Router(config-te-tun-autocapacity)# max-clones 4
Router(config-te-tun-autocapacity)# nominal-bandwidth 150
Router(config-te-tun-autocapacity)# split-bandwidth 200
Router(config-te-tun-autocapacity)# merge-bandwidth 100
```

Configuring Auto-Tunnel Backup

The MPLS Traffic Engineering Auto-Tunnel Backup feature enables a router to dynamically build backup tunnels on the interfaces that are configured with MPLS TE tunnels instead of building MPLS-TE tunnels statically.

The MPLS-TE Auto-Tunnel Backup feature has these benefits:

- Backup tunnels are built automatically, eliminating the need for users to pre-configure each backup tunnel and then assign the backup tunnel to the protected interface.
- Protection is expanded—FRR does not protect IP traffic that is not using the TE tunnel or Label Distribution Protocol (LDP) labels that are not using the TE tunnel.

The TE attribute-set template that specifies a set of TE tunnel attributes, is locally configured at the headend of auto-tunnels. The control plane triggers the automatic provisioning of a corresponding TE tunnel, whose characteristics are specified in the respective attribute-set.

Configuration Example

This example configures Auto-Tunnel backup on an interface and specifies the attribute-set template for the auto tunnels. In this example, unused backup tunnels are removed every 20 minutes using a timer and also the range of tunnel interface numbers are specified.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# auto-tunnel backup
Router(config-mpls-te-if-auto-backup)# attribute-set ab
Router(config-mpls-te)# auto-tunnel backup timers removal unused 20
Router(config-mpls-te)# auto-tunnel backup tunnel-id min 6000 max 6500
Router(config-mpls-te)# commit
```

Verification

This example shows a sample output for automatic backup tunnel configuration.

```
Router# show mpls traffic-eng tunnels brief

            TUNNEL NAME         DESTINATION      STATUS  STATE
             tunnel-te0           200.0.0.3          up  up
             tunnel-te1           200.0.0.3          up  up
             tunnel-te2           200.0.0.3          up  up
             tunnel-te50          200.0.0.3          up  up
           *tunnel-te60        200.0.0.3           up  up
              *tunnel-te70        200.0.0.3          up  up
              *tunnel-te80        200.0.0.3          up  up
```

Removing an AutoTunnel Backup

To remove all the backup autotunnels, perform this task.

Configuration Example

```
Router# clear mpls traffic-eng auto-tunnel backup unused all
```

Verification

Use the **show mpls traffic-eng auto-tunnel summary** command to verify MPLS-TE autotunnel information, including the ones removed.

Configuring Auto-Tunnel Mesh

The MPLS-TE auto-tunnel mesh (auto-mesh) feature allows you to set up full mesh of TE Point-to-Point (P2P) tunnels automatically with a minimal set of MPLS traffic engineering configurations. You can configure one or more mesh-groups and each mesh-group requires a destination-list (IPv4 prefix-list) listing destinations, which are used as destinations for creating tunnels for that mesh-group.

You can configure MPLS-TE auto-mesh type attribute-sets (templates) and associate them to mesh-groups. Label Switching Routers (LSRs) can create tunnels using the tunnel properties defined in this attribute-set.

Auto-Tunnel mesh configuration minimizes the initial configuration of the network. You can configure tunnel properties template and mesh-groups or destination-lists on TE LSRs that further creates full mesh of TE tunnels between those LSRs. It eliminates the need to reconfigure each existing TE LSR in order to establish a full mesh of TE tunnels whenever a new TE LSR is added in the network.

Configuration Example

This example configures an auto-tunnel mesh group and specifies the attributes for the tunnels in the mesh-group.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# auto-tunnel mesh
Router(config-mpls-te-auto-mesh)# tunnel-id min 1000 max 2000
Router(config-mpls-te-auto-mesh)# group 10
Router(config-mpls-te-auto-mesh-group)# attribute-set 10
Router(config-mpls-te-auto-mesh-group)# destination-list dl-65
Router(config-mpls-te)# attribute-set auto-mesh 10
Router(config-mpls-te-attribute-set)# autoroute announce
```

```
Router(config-mpls-te-attribute-set)# auto-bw collect-bw-only
Router(config)# commit
```

Verification

Verify the auto-tunnel mesh configuration using the **show mpls traffic-eng auto-tunnel mesh** command.

```
Router# show mpls traffic-eng auto-tunnel mesh

Auto-tunnel Mesh Global Configuration:
  Unused removal timeout: 1h 0m 0s
  Configured tunnel number range: 1000-2000

Auto-tunnel Mesh Groups Summary:
  Mesh Groups count: 1
  Mesh Groups Destinations count: 3
  Mesh Groups Tunnels count:
    3 created, 3 up, 0 down, 0 FRR enabled

Mesh Group: 10 (3 Destinations)
  Status: Enabled
  Attribute-set: 10
  Destination-list: dl-65 (Not a prefix-list)
  Recreate timer: Not running
       Destination    Tunnel ID    State  Unused timer
  ----------------  -----------  -------  ------------
       192.168.0.2         1000     up   Not running
       192.168.0.3         1001     up   Not running
       192.168.0.4         1002     up   Not running
  Displayed 3 tunnels, 3 up, 0 down, 0 FRR enabled

Auto-mesh Cumulative Counters:
  Last cleared: Wed Oct  3 12:56:37 2015 (02:39:07 ago)
                      Total
  Created:                3
  Connected:              0
  Removed (unused):       0
  Removed (in use):       0
  Range exceeded:         0
```

Configuring Fast Reroute

Fast reroute (FRR) provides link protection to LSPs enabling the traffic carried by LSPs that encounter a failed link to be rerouted around the failure. The reroute decision is controlled locally by the router connected to the failed link. The headend router on the tunnel is notified of the link failure through IGP or through RSVP. When it is notified of a link failure, the headend router attempts to establish a new LSP that bypasses the failure. This provides a path to reestablish links that fail, providing protection to data transfer. The path of the backup tunnel can be an IP explicit path, a dynamically calculated path, or a semi-dynamic path. For detailed conceptual information on fast reroute, see the MPLS-TE Features - Details topic.

Before You Begin

The following prerequisites are required to create an MPLS-TE tunnel:

- You must have a router ID for the neighboring router.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.

Configuration Example

This example configures fast reroute on an MPLS-TE tunnel. Here, tunnel-te 2 is configured as the back-up tunnel. You can use the **protected-by** command to configure path protection for an explicit path that is protected by another path.

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# fast-reroute
Router(config-if)# exit
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# backup-path tunnel-te 2
Router(config)# interface tunnel-te 2
Router(config-if)# backup-bw global-pool 5000
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# destination 192.168.92.125
Router(config-if)# path-option l explicit name backup-path protected by 10
Router(config-if)# path-option l0 dynamic
Router(config)# commit
```

Verification

Use the **show mpls traffic-eng fast-reroute database** command to verify the fast reroute configuration.

```
Router# show mpls traffic-eng fast-reroute database

Tunnel head FRR information:
Tunnel      Out intf/label                    FRR intf/label      Status
----------  ----------------                  ----------------    -------
tt4000       HundredGigabitEthernet 0/0/0/3:34     tt1000:34        Ready
tt4001       HundredGigabitEthernet 0/0/0/3:35     tt1001:35        Ready
tt4002       HundredGigabitEthernet 0/0/0/3:36     tt1001:36        Ready
```

Configuring Flexible Name-Based Tunnel Constraints

MPLS-TE Flexible Name-based Tunnel Constraints provides a simplified and more flexible means of configuring link attributes and path affinities to compute paths for the MPLS-TE tunnels.

In traditional TE, links are configured with attribute-flags that are flooded with TE link-state parameters using Interior Gateway Protocols (IGPs), such as Open Shortest Path First (OSPF).

MPLS-TE Flexible Name-based Tunnel Constraints lets you assign, or map, up to 32 color names for affinity and attribute-flag attributes instead of 32-bit hexadecimal numbers. After mappings are defined, the attributes can be referred to by the corresponding color name.

Configuration Example

This example shows assigning a how to associate a tunnel with affinity constraints.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# affinity-map red 1
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# attribute-names red
Router(config)# interface tunnel-te 2
Router(config-if)# affinity include red
Router(config)# commit
```

Configuring Forwarding Path

Perform this task to configure forwarding path in the MPLS-TE interface.

Configuration Example

```
Router # configure
Router(config)# interface tunnel-te 1
Router(config-if)# forward-class 1
Router(config-if)# exit
Router(config)# commit
```

Configuring an IETF DS-TE Tunnel Using MAM

IETF DS-TE mode uses IETF-defined extensions for RSVP and IGP. This mode interoperates with third-party vendor equipment. IETF mode supports multiple bandwidth constraint models, including Russian Doll Model (RDM) and Maximum Allocation Model (MAM), both with two bandwidth pools.

Configuration Example

This example configures an IETF DS-TE tunnel using MAM.

```
Router# configure
Router(config)# rsvp interface HundredGigabitEthernet 0/0/0/3
Router(config-rsvp-if)# bandwidth mam max-reservable-bw 1000 bc0 600 bc1 400
Router(config-rsvp-if)# exit
Router(config)# mpls traffic-eng
Router(config-mpls-te)# ds-te mode ietf
Router(config-mpls-te)# ds-te bc-model mam
Router(config-mpls-te)# exit
Router(config)# interface tunnel-te 2
Router(config-if)# signalled bandwidth sub-pool 10
Router(config)# commit
```

Verification

Use the **show mpls traffic-eng topology** command to verify the IETF DS-TE tunnel using MAM configuration.

Configuring an IETF DS-TE Tunnel Using RDM

IETF DS-TE mode uses IETF-defined extensions for RSVP and IGP. This mode interoperates with third-party vendor equipment.

IETF mode supports multiple bandwidth constraint models, including Russian Doll Model (RDM) and Maximum Allocation Model (MAM), both with two bandwidth pools. In an IETF DS-TE network, identical bandwidth constraint models must be configured on all nodes.

Before you Begin

The following prerequisites are required to create a IETF mode DS-TE tunnel using RDM:

- You must have a router ID for the neighboring router.

- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.

Configuration Example

This example configures an IETF DS-TE tunnel using RDM.

```
Router# configure
Router(config)# rsvp interface HundredGigabitEthernet 0/0/0/3
Router(config-rsvp-if)# bandwidth rdm 100 150
Router(config-rsvp-if)# exit
Router(config)# mpls traffic-eng
Router(config-mpls-te)# ds-te mode ietf
Router(config-mpls-te)# exit
Router(config)# interface tunnel-te 2
Router(config-if)# signalled bandwidth sub-pool 10 class-type 1
Router(config)# commit
```

Verification

Use the **show mpls traffic-eng topology** command to verify the IETF DS-TE tunnel using RDM configuration.

Configuring an MPLS Traffic Engineering Interarea Tunneling

The MPLS TE Interarea Tunneling feature allows you to establish MPLS TE tunnels that span multiple Interior Gateway Protocol (IGP) areas and levels. This feature removes the restriction that required the tunnel headend and tailend routers both to be in the same area. The IGP can be either Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF).To configure an inter-area tunnel, you specify on the headend router a loosely routed explicit path for the tunnel label switched path (LSP) that identifies each area border router (ABR) the LSP should traverse using the next-address loose command. The headend router and the ABRs along the specified explicit path expand the loose hops, each computing the path segment to the next ABR or tunnel destination.

Configuration Example

This example configures an IPv4 explicit path with ABR configured as loose address on the headend router.

```
Router# configure
Router(config)# explicit-path name interarea1
Router(config-expl-path)# index1 next-address loose ipv4 unicast 172.16.255.129
Router(config-expl-path)# index 2 next-address loose ipv4 unicast 172.16.255.131
Router(config)# interface tunnel-te1
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# destination 172.16.255.2
Router(config-if)# path-option 10 explicit name interarea1
Router(config)# commit
```

Configuring MPLS-TE Path Protection

Path protection provides an end-to-end failure recovery mechanism for MPLS-TE tunnels. A secondary Label Switched Path (LSP) is established, in advance, to provide failure protection for the protected LSP that is carrying a tunnel's TE traffic. When there is a failure on the protected LSP, the source router immediately enables the secondary LSP to temporarily carry the tunnel's traffic. Failover is triggered by a RSVP error message sent to the LSP head end. Once the head end received this error message, it switches over to the

secondary tunnel. If there is a failure on the secondary LSP, the tunnel no longer has path protection until the failure along the secondary path is cleared. Path protection can be used within a single area (OSPF or IS-IS), external BGP [eBGP], and static routes. Both the explicit and dynamic path-options are supported for the MPLS-TE path protection feature. You should make sure that the same attributes or bandwidth requirements are configured on the protected option.

Before You Begin

The following prerequisites are required for enabling path protection.

- You should ensure that your network supports MPLS-TE, Cisco Express Forwarding, and Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF).
- You should configure MPLS-TE on the routers.

Configuration Example

This example configures how to configure path protection for a mpls-te tunnel. The primary path-option should be present to configure path protection. In this configuration, R1 is the headend router and R3 is the tailend router for the tunnel while R2 and R4 are mid-point routers. In this example, 6 explicit paths and 1 dynamic path is created for path protection. You can have upto 8 path protection options for a primary path.

Note **Path-protection** through user-specified path-options is not supported and the **protected-by** is used specifically only for numbered tunnels and unavailable for named-tunnels.

```
Router # configure
Router(config)# interface tunnel-te 0
Router(config-if)# destination 192.168.3.3
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# autoroute announce
Router(config-if)# path-protection
Router(config-if)# path-option 1 explicit name r1-r2-r3-00 protected-by 2
Router(config-if)# path-option 2 explicit name r1-r2-r3-01 protected-by 3
Router(config-if)# path-option 3 explicit name r1-r4-r3-01 protected-by 4
Router(config-if)# path-option 4 explicit name r1-r3-00 protected-by 5
Router(config-if)# path-option 5 explicit name r1-r2-r4-r3-00 protected-by 6
Router(config-if)# path-option 6 explicit name r1-r4-r2-r3-00 protected-by 7
Router(config-if)# path-option 7 dynamic
Router(config-if)# exit
Router(config)# commit
```

Verification

Use the **show mpls traffic-eng tunnels** command to verify the MPLS-TE path protection configuration.

```
Router# show mpls traffic-eng tunnels 0

Name: tunnel-te0  Destination: 192.168.92.125  Ifhandle:0x8007d34
  Signalled-Name: router
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected
    path option 1,  type explicit r1-r2-r3-00 (Basis for Setup, path weight 2)
      Protected-by PO index: 2
    path option 2,  type explicit r1-r2-r3-01 (Basis for Standby, path weight 2)
      Protected-by PO index: 3
    path option 3,  type explicit r1-r4-r3-01
```

```
      Protected-by PO index: 4
    path option 4,  type explicit r1-r3-00
      Protected-by PO index: 5
    path option 5,  type explicit r1-r2-r4-r3-00
      Protected-by PO index: 6
    path option 6,  type explicit r1-r4-r2-r3-00
      Protected-by PO index: 7
    path option 7,  type dynamic
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Fri Oct 13 15:05:28 2017 (01:19:11 ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Delay-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (not enabled)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  History:
    Tunnel has been up for: 01:14:13 (since Fri Oct 13 15:10:26 UTC 2017)
    Current LSP:
      Uptime: 01:14:13 (since Fri Oct 13 15:10:26 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, identical to the [CURRENT] LSP
        Date/Time: Fri Oct 13 15:08:41 UTC 2017 [01:15:58 ago]
    Standby Reopt LSP:
      Last Failure:
        LSP not signalled, identical to the [STANDBY] LSP
        Date/Time: Fri Oct 13 15:08:41 UTC 2017 [01:15:58 ago]
        First Destination Failed: 192.3.3.3
    Prior LSP:
      ID: 8 Path Option: 1
      Removal Trigger: path protection switchover
    Standby LSP:
      Uptime: 01:13:56 (since Fri Oct 13 15:10:43 UTC 2017)
  Path info (OSPF 1 area 0):
  Node hop count: 2
  Hop0: 192.168.1.2
  Hop1: 192.168.3.1
  Hop2: 192.168.3.2
  Hop3: 192.168.3.3
  Standby LSP Path info (OSPF 1 area 0), Oper State: Up :
  Node hop count: 2
  Hop0: 192.168.2.2
  Hop1: 192.168.3.1
  Hop2: 192.168.3.2
  Hop3: 192.168.3.3
Displayed 1 (of 4001) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

Configuring Next Hop Backup Tunnel

The backup tunnels that bypass only a single link of the LSP path are referred as Next Hop (NHOP) backup tunnels because they terminate at the LSP's next hop beyond the point of failure. They protect LSPs, if a link along their path fails, by rerouting the LSP traffic to the next hop, thus bypassing the failed link.

Configuration Example

This example configures next hop backup tunnel on an interface and specifies the attribute-set template for the auto tunnels. In this example, unused backup tunnels are removed every 20 minutes using a timer and also the range of tunnel interface numbers are specified.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# auto-tunnel backup nhop-only
Router(config-mpls-te-if-auto-backup)# attribute-set ab
Router(config-mpls-te)# auto-tunnel backup timers removal unused 20
Router(config-mpls-te)# auto-tunnel backup tunnel-id min 6000 max 6500
Router(config)# commit
```

Configuring Point-to-Multipoint TE Tunnels

For P2MP tunnels, a Cisco 8000 Series router supports the mid-point router function, and does not support source or receiver functions. To know how to configure a source or receiver (destination) router in a P2MP tunnel, refer the MPLS configuration guide for the corresponding platform.

Configuring Point-to-Multipoint TE Auto-Tunnels

The P2MP-TE Auto-tunnels feature enables dynamic creation and management of P2MP auto-tunnels for the transport of VPLS traffic on Cisco IOS XR Software. The P2MP-TE auto-tunnel configuration is disabled by default. Use the **auto-tunnel p2mp tunnel-id** command to enable a P2MP-TE Auto-tunnel. This configures the tunnel ID range that can be allocated to P2MP auto-tunnels. This also determines the maximum number of P2MP auto-tunnels that can be created.

Configuration Example

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# auto-tunnel p2mp
Router(config-te-auto-p2mp)# tunnel-id min 10000 max 11000
Router(config-te-auto-p2mp)# commit
```

Enabling Soft-Preemption

Enabling Soft-Preemption on a Node

Perform this task to enable the soft-preemption feature in the MPLS TE configuration mode. By default, this feature is disabled. You can configure the soft-preemption feature for each node. It has to be explicitly enabled for each node.

Configuration Example

If soft-preemption is enabled, the head-end node tracks whether an LSP desires the soft-preemption treatment. However, when a soft-preemption feature is disabled on a node, this node continues to track all LSPs desiring soft-preemption. This is needed in a case when soft-preemption is re-enabled, TE will have the property of the existing LSPs without any re-signaling.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# soft-preemption
Router(config-soft-preemption)# timeout 100
Router(config-soft-preemption)# commit
```

Enabling Soft-Preemption on a Tunnel

Perform this task to enable the soft-preemption feature on a MPLS TE tunnel. By default, this feature is disabled. It has to be explicitly enabled.

Configuration Example

When soft preemption is enabled on a tunnel, a path-modify message is sent for the current LSP, reopt LSP, path protection LSP, and current LSP in FRR active state, with the **soft preemption desired** property.

```
Router# configure
Router(config)# interface tunnel-te 10
Router(config-if)# soft-preemption
Router(config-if)# commit
```

Enabling Soft-preemption over FRR Backup Tunnels

Before enabling soft-preemption over FRR backup, ensure that you enable soft-preemption, and activate the FRR backup tunnel.

Configuration Example

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# soft-preemption frr-rewrite
Router(config-mpls-te)# commit
```

Configuring Pre-Standard DS-TE

Regular traffic engineering does not provide bandwidth guarantees to different traffic classes. A single bandwidth constraint is used in regular TE that is shared by all traffic. MPLS DS-TE enables you to configure multiple bandwidth constraints on an MPLS-enabled interface. These bandwidth constraints can be treated differently based on the requirement for the traffic class using that constraint. Cisco IOS XR software supports two DS-TE modes: Pre-standard and IETF. Pre-standard DS-TE uses the Cisco proprietary mechanisms for RSVP signaling and IGP advertisements. This DS-TE mode does not interoperate with third-party vendor equipment. Pre-standard DS-TE is enabled only after configuring the sub-pool bandwidth values on MPLS-enabled interfaces.

Pre-standard Diff-Serve TE mode supports a single bandwidth constraint model a Russian Doll Model (RDM) with two bandwidth pools: global-pool and sub-pool.

Before You Begin

The following prerequisites are required to configure a Pre-standard DS-TE tunnel.

- You must have a router ID for the neighboring router.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.

Configuration Example

This example configures a pre-standard DS-TE tunnel.

```
Router# configure
Router(config)# rsvp interface HundredGigabitEthernet 0/0/0/3
Router(config-rsvp-if)# bandwidth 100 150 sub-pool 50
Router(config-rsvp-if)# exit
Router(config)# interface tunnel-te 2
Router(config-if)# signalled bandwidth sub-pool 10
Router(config)# commit
```

Verification

Use the **show mpls traffic-eng topology** command to verify the pre-standard DS-TE tunnel configuration.

Configuring SRLG Node Protection

Shared Risk Link Groups (SRLG) in MPLS traffic engineering refer to situations in which links in a network share common resources. These links have a shared risk, and that is when one link fails, other links in the group might fail too.

OSPF and IS-IS flood the SRLG value information (including other TE link attributes such as bandwidth availability and affinity) using a sub-type length value (sub-TLV), so that all routers in the network have the SRLG information for each link.

MPLS-TE SRLG feature enhances backup tunnel path selection by avoiding using links that are in the same SRLG as the interfaces it is protecting while creating backup tunnels.

Configuration Example

This example creates a backup tunnel and excludes the protected node IP address from the explicit path.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# backup-path tunnl-te 2
Router(config-mpls-te-if)# exit
Router(config)# interface tunnel-te 2
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# path-option 1 explicit name backup-srlg
Router(config-if)# destination 192.168.92.125
Router(config-if)# exit
Router(config)# explicit-path name backup-srlg-nodep
Router(config-if)# index 1 exclude-address 192.168.91.1
Router(config-if)# index 1 exclude-srlg 192.168.92.2
Router(config)# commit
```

SRLG Limitations

There are few limitations to the configured SRLG feature:

- The **exclude-address** and **exclude-srlg** options are not allowed in the IP **explicit path strict-address** network.
- Whenever SRLG values are modified after tunnels are signaled, they are verified dynamically in the next path verification cycle.

Creating an MPLS-TE Tunnel

Creating an MPLS-TE tunnel is a process of customizing the traffic engineering to fit your network topology. The MPLS-TE tunnel is created at the headend router. You need to specify the destination and path of the TE LSP.

To steer traffic through the tunnel, you can use the following ways:

- Static Routing
- Autoroute Announce
- Forwarding Adjacency

Before You Begin

The following prerequisites are required to create an MPLS-TE tunnel:

- You must have a router ID for the neighboring router.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.

Configuration Example

This example configures an MPLS-TE tunnel on the headend router with a destination IP address 192.168.92.125. The bandwidth for the tunnel, path-option, and forwarding parameters of the tunnel are also configured. You can use static routing, autoroute announce or forwarding adjacency to steer traffic through the tunnel.

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# destination 192.168.92.125
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# path-option 1 dynamic

Router(config-if)# autoroute announce | forwarding-adjacency
Router(config-if)# signalled-bandwidth 100
Router(config)# commit
```

Verification

Verify the configuration of MPLS-TE tunnel using the following command.

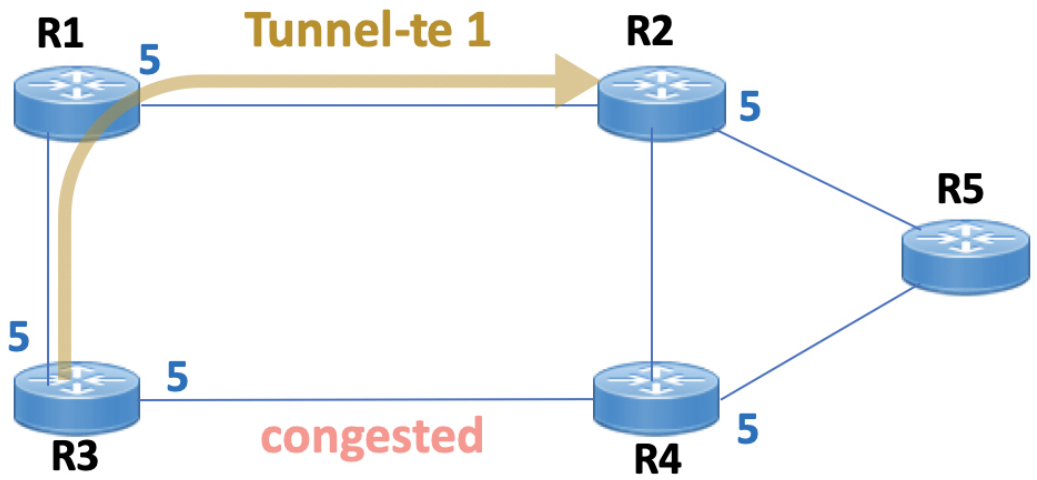
```
Router# show mpls traffic-engineering tunnels brief

            Signalling Summary:
                LSP Tunnels Process:  running
                       RSVP Process:  running
                         Forwarding:  enabled
```

```
Periodic reoptimization:  every 3600 seconds, next in 2538 seconds
 Periodic FRR Promotion:  every 300 seconds, next in 38 seconds
Auto-bw enabled tunnels:  0 (disabled)
           TUNNEL NAME          DESTINATION      STATUS  STATE


          tunnel-te1       192.168.92.125      up       up
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

Automatic Modification Of An MPLS-TE Tunnel's Metric

If the IGP calculation on a router results in an equal cost multipath (ECMP) scenario where next-hop interfaces are a mix of MPLS-TE tunnels and physical interfaces, you may want to ensure that a TE tunnel is preferred. Consider this topology:

Figure 13: MPLS-TE Tunnel



1. All links in the network have a metric of 5.
2. To offload a congested link between R3 and R4, an MPLS-TE tunnel is created from R3 to R2.
3. If the metric of the tunnel is also 5, traffic from R3 to R5 is load-balanced between the tunnel and the physical R3-R4 link.

To ensure that the MPLS-TE tunnel is preferred in such scenarios, configure the **autoroute metric** command on the tunnel interface. The modified metric is applied in the routing information base (RIB), and the tunnel is preferred over the physical path of the same metric. Sample configuration:

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# autoroute metric relative -1
```

The **autoroute metric** command syntax is **autoroute metric** {**absolute**|**relative**} *value*

- **absolute** enables the absolute metric mode, for a metric range between 1 and 2147483647.
- **relative** enables the relative metric mode, for a metric range between -10 and 10, including zero.

Note

- Since the **relative** metric is not saved in the IGP database, the advertised metric of the MPLS-TE tunnel remains 5, and doesn't affect SPF calculation outcomes on other nodes.
- Configuring Segment Routing and Autoroute Destination together is not supported. If autoroute functionality is required in an Segment Routing network, we recommend you to configure Autoroute Announce.

Configuring Dark Bandwidth Accounting

To enable RSVP-TE Dark Bandwidth Accounting feature, perform the following steps:

1. Enable per-interface aggregate SR counters.
2. Configure TE dark bandwidth accounting.

SUMMARY STEPS

1. **configure**
2. **hw-module profile cef dark-bw enable**
3. **mpls traffic-eng**
4. **bandwidth-accounting**
5. **application interval** *seconds*
6. **application enforced**
7. **sampling-interval** *seconds*
8. **adjustment-factor** *percentage*
9. **flooding threshold up** *percentage* **down** *percentage*
10. Use the **commit** or **end** command.
11. **mpls traffic-eng link-management bandwidth-accounting enforce all**
12. **clear mpls traffic-eng link-management bandwidth-accounting**
13. **show interface** *type_path* **accounting**
14. **show mpls traffic-eng link-management summary**
15. **show mpls traffic-eng link-management advertisements**
16. **show mpls traffic-eng link-management interfaces** [*type interface-path-id*] [**detail**] [**bandwidth-accounting**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**hw-module profile cef dark-bw enable** **Example:**	Enables per-interface aggregate SR counters for all interfaces on the router.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# hw-module profile cef dark-bandwidth enable` `RP/0/RP0/CPU0:router(config)# commit`	**Note** After you enter this command, you must reload the router. **Caution** This command should only be enabled on a router where a prefix with an SR prefix SID learned via ECMP has the same out label across all its paths. This condition is met for prefixes learned via ECMP in an SR network with homogenous SRGB and when either no protection or IP-FRR LFA protection is enabled. Do not use this command on a router with TI-LFA enabled while expecting backup paths that would require extra labels to be imposed. In Cisco IOS XR release 7.3.3 and earlier, do not use this command on a router where a prefix with an SR prefix SID is learned via ECMPs with different egress action (pop and swap). Label programming errors and traffic loss would be observed for those prefixes. In Cisco IOS XR release 7.3.4 and later, this restriction no longer applies.
Step 3	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`	Enters MPLS TE configuration mode.
Step 4	**bandwidth-accounting** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# bandwidth-accounting`	Enables RSVP-TE dark bandwidth accounting and enters bandwidth accounting configuration mode.
Step 5	**application interval** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# application interval 90`	Configures the length of the application interval in seconds. At the end of application interval, dark bandwidth rates are computed and applied to all RSVP-TE enabled interfaces. If the interval is reconfigured while the timer is running, the new value is compared to the time remaining for the running timer. The timer is adjusted so that the lower of these two values is used for this interval. The subsequent interval will use the newly configured value. **Note** TE stores sample history for the current and previous application intervals. If the application interval is lowered, TE may discard the sample history.

	Command or Action	Purpose
		Range is from 90 to 1800. The default value is 180.
Step 6	**application enforced** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# application enforced`	Enables enforcement of the calculated BMRe rate.
Step 7	**sampling-interval** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# sampling-interval 30`	Configures the length of the sampling interval in seconds. The dark bandwidth rate is collected from the statistics collector process (statsD) at the end of each sampling interval for each TE link. If the interval is reconfigured while the timer is running, the new value is compared to the time remaining for the running timer. The timer is adjusted so that the lower of these two values is used for this interval. The subsequent interval will use the newly configured value. Range is from 10 to 600. The default is 60.
Step 8	**adjustment-factor** *percentage* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# adjustment-factor 200`	Configures TE to over-book (>100%) or under-book (<100%) the effective maximum reservable bandwidth (BMRe). The measured dark-bandwidth will be scaled based on the adjustment factor. Range is from 0 to 200. The default value is 100.
Step 9	**flooding threshold up** *percentage* **down** *percentage* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# flooding threshold up 30 down 30`	Configures the reserved bandwidth thresholds. When bandwidth crosses one of these thresholds, flooding is triggered. Range is from 0 to 100. The default value is 10.
Step 10	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 11	**mpls traffic-eng link-management bandwidth-accounting enforce all**	(Optional)

	Command or Action	Purpose
	Example: `RP/0/RP0/CPU0:router# mpls traffic-eng link-management bandwidth-accounting enforce all`	Applies the measured rates immediately. When you apply measured rates immediately, the RSVP-TE bandwidth-accounting might flood the updated bandwidth values immediately. Applying measured rates immediately does not affect the periodic application of the bandwidth.
Step 12	**clear mpls traffic-eng link-management bandwidth-accounting** **Example:** `RP/0/RP0/CPU0:router# clear mpls traffic-eng link-management bandwidth-accounting`	(Optional) Erases the collected sample history and resets the application and sample timers.
Step 13	**show interface** *type_path* **accounting** **Example:** `RP/0/RP0/CPU0:router# show interface hundredGigE 0/0/0/26 accounting`	(Optional) Displays the per-interface SR accounting.
Step 14	**show mpls traffic-eng link-management summary** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng link-management summary`	(Optional) Displays a summary of link management information, including bandwidth accounting information.
Step 15	**show mpls traffic-eng link-management advertisements** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng link-management advertisements`	(Optional) Displays local link information that MPLS-TE link management is currently flooding into the global TE topology.
Step 16	**show mpls traffic-eng link-management interfaces** [*type interface-path-id*] [**detail**] [**bandwidth-accounting**] **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng link-management interfaces gig0/1/1/1 detail`	(Optional) Displays bandwidth accounting and utilization details and link management information.

To display the per-interface SR counters, use the **show interface** *type_path* **accounting** command:

```
RP/0/RP0/CPU0:router# show interface hundredGigE 0/0/0/26 accounting
Mon Feb  3 23:29:48.449 UTC
HundredGigE0/0/0/26
  Protocol              Pkts In         Chars In     Pkts Out        Chars Out
  ARP                         3              222            3              126
  IPV6_ND                    11             1122           13             1112
```

```
CLNS                      99          121910          94          116212
SR_MPLS                    0               0        3126          581436
```

Note The SR_MPLS counter is an egress-only counter and includes all traffic from the following:

- IPv4 unlabelled - SR last-hop traffic after PHP
- IPv6 unlabelled - SR last-hop traffic after PHP
- SR label switched traffic

To display detailed SR bandwidth utilization, use the **show mpls traffic-eng link-management interface** *type_path* **detail** command:

```
Router# show mpls traffic-eng link-management interface hundredGigE 0/0/0/26 detail
bandwidth-accounting

  System Information::
      Links Count         : 16 (Maximum Links Supported 800)

  Link ID:: HundredGigE0/0/0/26 (26.1.1.1)
    Local Intf ID: 22
    Link Status:

      Link Label Type          : PSC
      Physical BW              : 1000000 kbits/sec
      BCID                     : RDM
      Max Reservable BW        : 529309 kbits/sec (reserved: 94% in, 94% out)
      Flooded Max Reservable BW: 529309 kbits/sec
      BC0 (Res. Global BW)     : 529309 kbits/sec (reserved: 94% in, 94% out)
      BC1 (Res. Sub BW)        : 0 kbits/sec (reserved: 100% in, 100% out)
      MPLS TE Link State       : MPLS TE on, RSVP on, admin-up
      IGP Neighbor Count       : 1
      Max Res BW (RDM)     : 900000 kbits/sec
      BC0 (RDM)            : 900000 kbits/sec
      BC1 (RDM)            : 0 kbits/sec
      Max Res BW (MAM)     : 0 kbits/sec
      BC0 (MAM)            : 0 kbits/sec
      BC1 (MAM)            : 0 kbits/sec

    Bandwidth Accounting: Segment-Routing
    Bandwidth Accounting Enforced: Yes
      Bandwidth Utilization Details:
        Sampling Interval               : 30 sec
        Application Interval            : 90 sec
        Adjustment Factor               : 200%
        Max Reservable BW Up Threshold  : 30
        Max Reservable BW Down Threshold: 30

  Last Application at:  23:46:32 Mon 03 Feb 2020 (51 seconds ago)
          Segment-Routing BW Utilization      : 185346 kbits/sec
          Adjusted BW Utilization             : 370692 kbits/sec
          Enforced BW Utilization             : 370692 kbits/sec
        Next Application at:  19:42:43 Sun 30 Apr 2017 (in 38 seconds)
        Last Collection at :  19:41:42 Sun 30 Apr 2017 (23 seconds ago)
        Next Collection at :  19:42:11 Sun 30 Apr 2017 (in 6 seconds)

  Bandwidth Samples (Kbps):
        Timestamp                     Segment-Routing
```

```
      19:40:12 Sun 30 Apr 2017               187961
      19:40:42 Sun 30 Apr 2017               180130
      19:41:12 Sun 30 Apr 2017               187949
```

To display a summary of link management information, including bandwidth accounting information, use the **show mpls traffic-eng link-management summary** command:

```
Router# show mpls traffic-eng link-management summary

  System Information::
      Links Count         : 14 (Maximum Links Supported 800)
      Flooding System     : enabled
      IGP Areas Count     : 1

  IGP Areas
  ----------

  IGP Area[1]:: IS-IS 0 level 2
      Flooding Protocol   : IS-IS
      Flooding Status     : flooded
      Periodic Flooding   : enabled (every 180 seconds)
      Flooded Links       : 7
      IGP System ID       : 0000.0000.0001
      MPLS TE Router ID   : 10.0.0.1
      IGP Neighbors       : 7

  Bandwidth accounting:
      Sampling interval: 30 seconds, Next in 29 seconds
      Application interval: 90 seconds, Next in 1 seconds
```

To display local link information that MPLS-TE link management is currently flooding into the global TE topology, use the **showmpls traffic-eng link-management advertisements** command:

```
Router# show mpls traffic-eng link-management advertisements

  Flooding Status             : Ready
  Last Flooding               : 470 seconds ago
  Last Flooding Trigger       : Link BW changed
  Next Periodic Flooding In   : 143 seconds
  Diff-Serv TE Mode           : Not enabled
  Configured Areas            : 1

  IGP Area[1]:: IS-IS 0 level 2
      Flooding Protocol   : IS-IS
      IGP System ID       : 0000.0000.0001
      MPLS TE Router ID   : 10.0.0.1
      Flooded Links       : 5

      Link ID:: 0 (GigabitEthernet0/1/1/0)
          Link IP Address      : 10.12.110.1
          O/G Intf ID          : 22
          Neighbor             : ID 0000.0000.0002.00, IP 10.12.110.2
          TE Metric            : 10
          IGP Metric           : 10
          Physical BW          : 1000000 kbits/sec
          BCID                 : RDM
          Max Reservable BW    : 899999 kbits/sec
          Res Global BW        : 899999 kbits/sec
          Res Sub BW           : 0 kbits/sec
```

Configure Autoroute Tunnel as Designated Path

Table 11: Feature History Table

Feature Name	Release Information	Feature Description
Configure Autoroute Tunnel as Designated Path	Release 7.6.2	Simplify the path selection for a traffic class and split traffic among multiple TE tunnels to achieve many benefits such as security and service-level agreements. You can now exclusively specify an autoroute tunnel to forward traffic to a particular tunnel destination address without considering the IS-IS metric for traffic path selection. Earlier, MPLS-TE considered either the Forwarding Adjacency (FA) or Autoroute (AA) tunnel to forward traffic based only on IS-IS metric. The feature introduces the **mpls traffic-eng tunnel restricted** command.

MPLS-TE builds a unidirectional tunnel from a source to a destination using label switched path (LSP) to forward traffic.

To forward the traffic through MPLS tunneling, you can use autoroute, forwarding adjacency, or static routing:

- Autoroute (AA) functionality allows to insert the MPLS TE tunnel in the Shortest Path First (SPF) tree for the tunnel to transport all the traffic from the headend to all destinations behind the tail-end. AA is only known to the tunnel headend router.
- Forwarding Adjacency (FA) allows the MPLS-TE tunnel to be advertised as a link in an IGP network with the cost of the link associated with it. Routers outside of the TE domain can see the TE tunnel and use it to compute the shortest path for routing traffic throughout the network.
- Static routing allows you to inject static IP traffic into a tunnel as the output interface for the routing decision.

Prior to this release, by default, MPLS-TE considers FA or AA tunnels to forward traffic based on the IS-IS metric. The lower metric is always used to forward traffic. There was no mechanism to forward traffic to a specific tunnel interface.

For certain prefixes to achieve many benefits such as security and service-level agreements, there might be a need to forward traffic to a specific tunnel interface that has a matching destination address.

With this feature, you can exclusively use AA tunnels to forward traffic to their tunnel destination address irrespective of IS-IS metric. Traffic steering is performed based on the prefixes and not metrics. Traffic to other prefixes defaults to the forwarding-adjacency (FA) tunnels.

To enable this feature, use the **mpls traffic-eng tunnel restricted** command.

Also, you may require more than one AA tunnel to a particular remote PE and use ECMP to forward traffic across AA tunnels. You can configure a loopback interface with one primary address and multiple secondary addresses on the remote PE, using one IP for the FA tunnel destination, and others for the AA tunnels destinations. Multiple IP addresses are advertised in the MPLS TE domain using the typed length value (TLV) 132 in IS-IS. A TLV-encoded data stream contains code related to the record type, the record length of the value, and value. TLV 132 represents the IP addresses of the transmitting interface.

Feature Behavior

When MPLS-TE tunnel restricted is configured, the following is the behavior:

- A complete set of candidate paths is available for selection on a per-prefix basis during RIB update as the first hop computation includes all the AA tunnels terminating on a node up to a limit of 64 and the lowest cost forwarding-adjacency or native paths terminating on the node or inherited from the parent nodes in the first hops set for the node.
- During per-prefix computation, AA tunnel first hops are used for traffic sent to their tunnel destination address even if FA tunnel or native first hops have a better metric. AA tunnel first-hops are not used for any other prefixes.
- ECMP is used when multiple AA tunnel first hops have the same destination address and metric.
- During per-prefix computation, AA tunnel first hops are used for traffic sent to their tunnel destination address, and for all other destinations on the tunnel tail node or behind it, even if a native path has a better metric.

  Adding **mpls traffic-eng tunnel preferred** configuration has no effect when the tunnel restricted is already configured.
- If there's no AA tunnel or if the tunnel is down, then native paths are used for all other destinations on the tunnel tail node or behind it.

  The route metric for a prefix reflects the chosen first-hop, not necessarily the lowest cost SPF distance to the node.

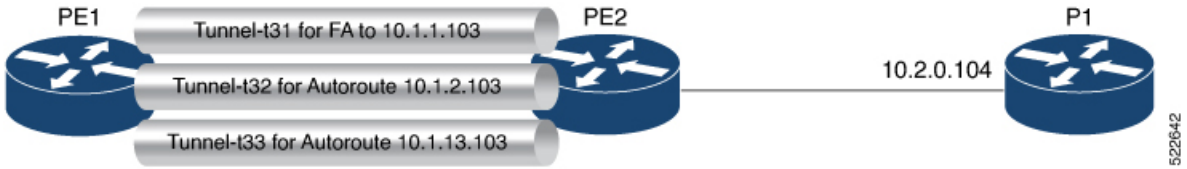
Restrictions for Configure Autoroute Tunnel as Designated Path

- The total number of interface addresses to the number that can be contained in 255 bytes is 63 for IPv4 and 15 for IPv6.
- When this feature is enabled, a maximum of 64 tunnels can terminate on the tail node.

Configure Autoroute Tunnel as Designated Path

Let's understand how to configure the feature using the following topology:

Figure 14: Topology



Consider the topology where PE1 has three MPLS tunnels connecting to PE2.

- Tunnel-t31: Forwarding adjacency (FA) is configured to the primary address of Loopback 0 on PE2 (10.1.1.103).
- Tunnel- t32: Autoroute announce (AA) is configured to a secondary address of Loopback 0 on PE2 (10.1.2.103).
- Tunnel-t33: Autoroute announce (AA) is configured to a secondary address of Loopback 0 on PE2 (10.1.3.103).

This feature is not enabled by default. When this feature is not enabled, traffic is load balanced over all AA tunnels towards the same remote PE provided the tunnel metric is the same:

```
Router# show routes
i L2 10.1.1.103/32 [115/40] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/40] via 10.1.3.103, 00:00:30, tunnel-t33
i L2 10.1.2.103/32 [115/40] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/40] via 10.1.3.103, 00:00:30, tunnel-t33
i L2 10.1.3.103/32 [115/40] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/40] via 10.1.3.103, 00:00:30, tunnel-t33
i L2 10.2.0.103/32 [115/40] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/40] via 10.1.3.103, 00:00:30, tunnel-t33
 10.2.0.104/32 [115/50] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/50] via 10.1.3.103, 00:00:30, tunnel-t33
```

Configuration Example

You can configure the feature using the **mpls traffic-eng tunnel restricted** command.

```
RP/0/RSP0/CPU0:ios# configure
RP/0/RSP0/CPU0:ios(config)# router isis 1
RP/0/RSP0/CPU0:ios(config-isis)# address-family ipv4 unicast
RP/0/RSP0/CPU0:ios(config-isis-af# mpls traffic-eng tunnel restricted
```

Running Configuration

The following example shows the AA tunnel metric running configuration:

```
router isis 1
 address-family ipv4 unicast
  mpls traffic-eng tunnel restricted
 !
!
end
```

Verification

When you enable the feature, traffic towards a particular prefix is sent only over the tunnel that has that IP address as destination.

```
Router# show route
i L2 10.1.1.103/32 [115/40] via 10.1.1.103, 00:00:04, tunnel-t31
i L2 10.1.2.103/32 [115/40] via 10.1.2.103, 00:00:04, tunnel-t32
i L2 10.1.3.103/32 [115/40] via 10.1.3.103, 00:00:04, tunnel-t33
i L2 10.2.0.103/32 [115/40] via 10.1.1.103, 00:00:04, tunnel-t31
i L2 10.2.0.104/32 [115/50] via 10.1.1.103, 00:00:04, tunnel-t31
```

When multiple restricted AA tunnels are created towards the same destination IP address, router load balances traffic across all those tunnels:

```
Router# show route
i L2 10.1.1.103/32 [115/40] via 10.1.1.101, 00:00:08, GigabitEthernet0/0/0/2
                   [115/40] via 10.1.3.101, 00:00:08, GigabitEthernet0/0/0/3
i L2 10.1.2.103/32 [115/40] via 10.1.2.103, 00:00:08, tunnel-t32
                   [115/40] via 10.1.2.103, 00:00:30, tunnel-t34
i L2 10.1.3.103/32 [115/40] via 10.1.3.103, 00:00:08, tunnel-t33
i L2 10.2.0.103/32 [115/40] via 10.1.1.101, 00:00:08, GigabitEthernet0/0/0/2
                   [115/40] via 10.1.3.101, 00:00:08, GigabitEthernet0/0/0/3
i L2 10.2.0.104/32 [115/50] via 10.1.1.101, 00:00:08, GigabitEthernet0/0/0/2
                   [115/50] via 10.1.3.101, 00:00:08, GigabitEthernet0/0/0/3
```

MPLS-TE Features - Details

MPLS TE Fast Reroute Link and Node Protection

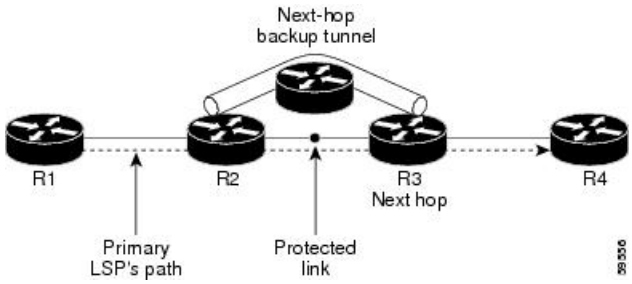
Fast Reroute (FRR) is a mechanism for protecting MPLS TE LSPs from link and node failures by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend routers try to establish new end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or node.

Note If FRR is greater than 50ms, it might lead to a loss of traffic.

Backup tunnels that bypass only a single link of the LSP's path provide link protection. They protect LSPs if a link along their path fails by rerouting the LSP's traffic to the next hop (bypassing the failed link). These tunnels are referred to as next-hop (NHOP) backup tunnels because they terminate at the LSP's next hop beyond the point of failure.

The following figure illustrates link protection.

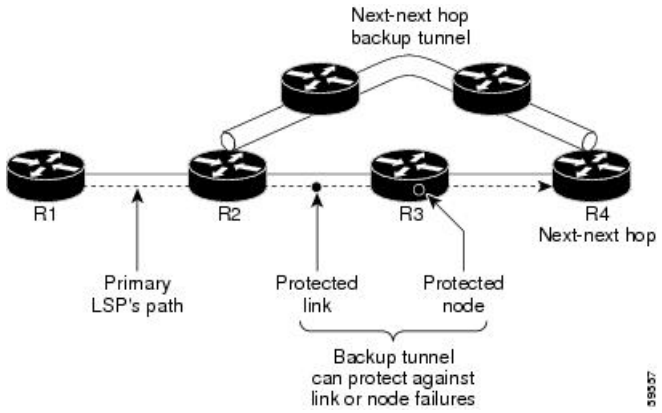
Figure 15: Link Protection



FRR provides node protection for LSPs. Backup tunnels that bypass next-hop nodes along LSP paths are called next-next-hop (NNHOP) backup tunnels because they terminate at the node following the next-hop node of the LSP paths, bypassing the next-hop node. They protect LSPs if a node along their path fails by enabling the node upstream of the failure to reroute the LSPs and their traffic around the failed node to the next-next hop. NNHOP backup tunnels also provide protection from link failures, because they bypass the failed link and the node.

The following figure illustrates node protection.

Figure 16: Node Protection



Differentiated Services Traffic Engineering

MPLS Differentiated Services Aware Traffic Engineering (DS-TE) is an extension of the regular MPLS-TE feature. Regular traffic engineering does not provide bandwidth guarantees to different traffic classes. A single bandwidth constraint is used in regular TE that is shared by all traffic. To support various classes of service (CoS), you can configure multiple bandwidth constraints. These bandwidth constraints can be treated differently based on the requirement for the traffic class using that constraint.

Cisco IOS XR software supports two DS-TE modes: pre-standard and IETF. The pre-standard DS-TE mode uses the Cisco proprietary mechanisms for RSVP signaling and IGP advertisements. This DS-TE mode does not interoperate with third-party vendor equipment. Pre-standard DS-TE is enabled only after configuring the sub-pool bandwidth values on MPLS-enabled interfaces. Pre-standard DS-TE mode supports a single bandwidth constraint model a Russian Doll Model (RDM) with two bandwidth pools: global-pool and sub-pool. TE class map is not used with Pre-standard DS-TE mode.

IETF DS-TE mode uses IETF-defined extensions for RSVP and IGP. This mode inter-operates with third-party vendor equipment. IETF mode supports multiple bandwidth constraint models, including RDM and Maximum Allocation Bandwidth Constraint Model (MAM), both with two bandwidth pools. In an IETF DS-TE network, identical bandwidth constraint models must be configured on all nodes. TE class map is used with IETF DS-TE mode and must be configured the same way on all nodes in the network.

The MAM constraint model has the following characteristics:

- Easy to use and intuitive.
- Isolation across class types.
- Simultaneously achieves isolation, bandwidth efficiency, and protection against QoS degradation.

The RDM constraint model has these characteristics:

- Allows greater sharing of bandwidth among different class types.
- Ensures bandwidth efficiency simultaneously and protection against QoS degradation of all class types.
- Specifies that it is used with preemption to simultaneously achieve isolation across class-types such that each class-type is guaranteed its share of bandwidth, bandwidth efficiency, and protection against QoS degradation of all class types.

MPLS-TE Forwarding Adjacency

MPLS TE forwarding adjacency allows you to handle a TE label-switched path (LSP) tunnel as a link in an Interior Gateway Protocol (IGP) network that is based on the Shortest Path First (SPF) algorithm. Both Intermediate System-to-Intermediate System (IS-IS) and Open Shortest Path First (OSPF) are supported as the IGP. A forwarding adjacency can be created between routers regardless of their location in the network. The routers can be located multiple hops from each other.

As a result, a TE tunnel is advertised as a link in an IGP network with the tunnel's cost associated with it. Routers outside of the TE domain see the TE tunnel and use it to compute the shortest path for routing traffic throughout the network. TE tunnel interfaces are advertised in the IGP network just like any other links. Routers can then use these advertisements in their IGPs to compute the SPF even if they are not the headend of any TE tunnels.

Automatic Bandwidth

Automatic bandwidth allows you to dynamically adjust bandwidth reservation based on measured traffic. MPLS-TE automatic bandwidth is configured on individual Label Switched Paths (LSPs) at every headend router. MPLS-TE automatic bandwidth monitors the traffic rate on a tunnel interface and resizes the bandwidth on the tunnel interface to align it closely with the traffic in the tunnel.

MPLS-TE automatic bandwidth can perform these functions:

- Monitors periodic polling of the tunnel output rate
- Resizes the tunnel bandwidth by adjusting the highest rate observed during a given period.

For every traffic-engineered tunnel that is configured for an automatic bandwidth, the average output rate is sampled, based on various configurable parameters. Then, the tunnel bandwidth is readjusted automatically based on either the largest average output rate that was noticed during a certain interval, or a configured maximum bandwidth value.

While re-optimizing the LSP with the new bandwidth, a new path request is generated. If the new bandwidth is not available, the last good LSP remains used. This way, the network experiences no traffic interruptions. If minimum or maximum bandwidth values are configured for a tunnel, the bandwidth, which the automatic bandwidth signals, stays within these values.

The output rate on a tunnel is collected at regular intervals that are configured by using the **application** command in MPLS-TE auto bandwidth interface configuration mode. When the application period timer expires, and when the difference between the measured and the current bandwidth exceeds the adjustment threshold, the tunnel is re-optimized. Then, the bandwidth samples are cleared to record the new largest output rate at the next interval. If a tunnel is shut down, and is later brought again, the adjusted bandwidth is lost, and the tunnel is brought back with the initially configured bandwidth. When the tunnel is brought back, the application period is reset.

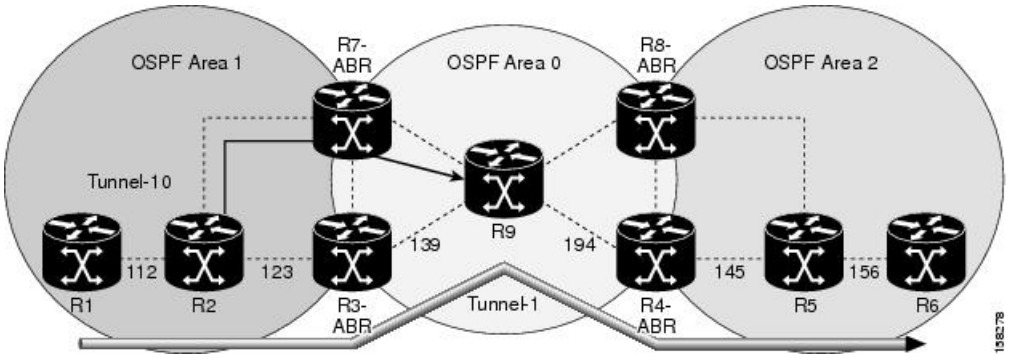
MPLS Traffic Engineering Interarea Tunneling

The MPLS-TE interarea tunneling feature allows you to establish TE tunnels spanning multiple Interior Gateway Protocol (IGP) areas and levels, thus eliminating the requirement that headend and tailend routers reside in a single area.

Interarea support allows the configuration of a TE LSP that spans multiple areas, where its headend and tailend label switched routers (LSRs) reside in different IGP areas. Customers running multiple IGP area backbones (primarily for scalability reasons) requires Multiarea and Interarea TE . This lets you limit the amount of flooded information, reduces the SPF duration, and lessens the impact of a link or node failure within an area, particularly with large WAN backbones split in multiple areas.

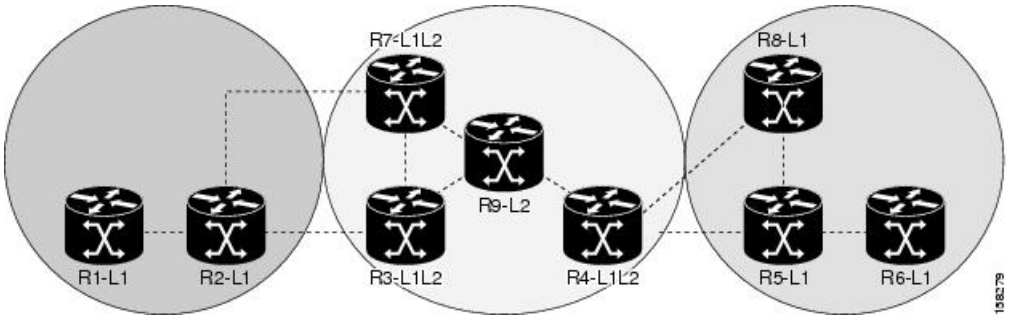
The following figure shows a typical interarea TE network using OSPF.

Figure 17: Interarea (OSPF) TE Network Diagram



The following figure shows a typical interlevel (IS-IS) TE Network.

Figure 18: Interlevel (IS-IS) TE Network Diagram



As shown in the topology, R2, R3, R7, and R4 maintain two databases for routing and TE information. For example, R3 has TE topology information related to R2, flooded through Level-1 IS-IS LSPs plus the TE topology information related to R4, R9, and R7, flooded as Level 2 IS-IS Link State PDUs (LSPs) (plus, its own IS-IS LSP).

Loose hop optimization allows the re-optimization of tunnels spanning multiple areas and solves the problem which occurs when an MPLS-TE LSP traverses hops that are not in the LSP's headend's OSPF area and IS-IS level. Interarea MPLS-TE allows you to configure an interarea traffic engineering (TE) label switched path (LSP) by specifying a loose source route of ABRs along the path. Then it is the responsibility of the ABR (having a complete view of both areas) to find a path obeying the TE LSP constraints within the next area to reach the next hop ABR (as specified on the headend router). The same operation is performed by the last ABR connected to the tailend area to reach the tailend LSR.

You must be aware of these considerations when using loose hop optimization:

- You must specify the router ID of the ABR node (as opposed to a link address on the ABR).
- When multiarea is deployed in a network that contains subareas, you must enable MPLS-TE in the subarea for TE to find a path when loose hop is specified.
- You must specify the reachable explicit path for the interarea tunnel.

Configuring Performance Measurement

Network performance metrics such as packet loss, delay, delay variation, and bandwidth utilization is a critical measure for traffic engineering (TE) in service provider networks. These network performance metrics provide network operators information about the performance characteristics of their networks for performance evaluation and helps to ensure compliance with service level agreements. The service-level agreements (SLAs) of service providers depend on the ability to measure and monitor these network performance metrics. Network operators can use performance measurement (PM) feature to monitor the network metrics for links as well as end-to-end TE label switched paths (LSPs).

Path Calculation Metric Type

To configure the metric type to be used for path calculation for a given tunnel, use the **path-selection metric** command in either the MPLS-TE configuration mode or under the tunnel interface configuration mode.

The metric type specified per interface takes the highest priority, followed by the MPLS-TE global metric type.

Note If the delay metric is configured, CSPF finds a path with optimized*minimum* link delay metric. See the *Configuring Performance Measurement* chapter in the Segment Routing Configuration Guide for information on configuring interface performance delay measurement.

Configuration Example

The following example shows how to set the path-selection metric to use the IGP metric under a specific tunnel interface:

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# path-selection metric igp
Router(config-if)# commit
```

The following example shows how to set the path-selection metric to use the delay metric under the MPLS-TE configuration mode:

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# path-selection metric delay
Router(config-mpls-te)# commit
```

Path-Selection Delay Limit

Apply the **path-selection delay-limit** configuration to set the upper limit on the path aggregate delay when computing paths for MPLS-TE LSPs. After you configure the **path-selection delay-limit** value, if the sum of minimum-delay metric from all links that are traversed by the path exceeds the specified delay-limit, CSPF will not return any path. The periodic path verification checks if the delay-limit is crossed.

The **path-selection delay-limit** value can be configured at the global MPLS-TE, per-interface tunnel, and per path-option attribute set. The path-selection delay-limit per path-option attribute set takes the highest priority, followed by per-interface, and then the MPLS-TE global path-selection delay-limit values.

The delay limit range is a value from 1 to 4294967295 microseconds.

Note See the *Configuring Performance Measurement* chapter in the Segment Routing Configuration Guide for information on configuring interface performance delay measurement.

Configuration Example

The following example shows how to set the path-selection delay limit under a specific tunnel interface:

```
Router# configure
Router(config)# interface tunnel-te2000
Router(config-if)# path-selection metric delay
Router(config-if)# path-selection delay-limit 200
Router(config-if)# commit
```

The following example shows how to set the path-selection delay limit under a path-option attribute set:

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# attribute-set path-option test
Router(config-te-attribute-set)# path-selection delay-limit 300
Router(config-te-attribute-set)# root
Router(config)# interface tunnel-te1000
Router(config-if)# path-option 10 dynamic attribute-set test
Router(config-if)# commit
```

The following example shows how to set the path-selection delay limit under the global MPLS-TE configuration mode:

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# path-selection metric delay
Router(config-mpls-te)# path-selection delay-limit 150
Router(config-mpls-te)# commit
```

Additional References

For additional information related to implementing MPLS-TE, refer to the following references:

Related Documents

Related Topic	Document Title
MPLS-TE commands	*MPLS Traffic Engineering Commands* module in *MPLS Command Reference for Cisco 8000 Series Routers*.

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

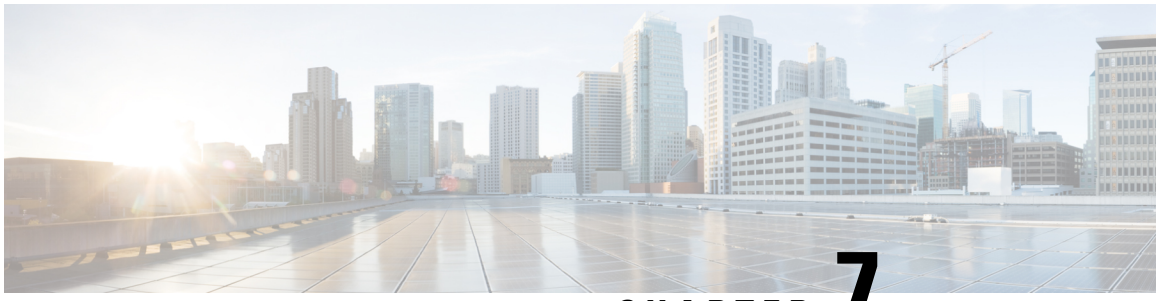
MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 4124	*Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering*, F. Le Faucheur, Ed. June 2005. (Format: TXT=79265 bytes) (Status: PROPOSED STANDARD)
RFC 4125	*Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*, F. Le Faucheur, W. Lai. June 2005. (Format: TXT=22585 bytes) (Status: EXPERIMENTAL)
RFC 4127	*Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*, F. Le Faucheur, Ed. June 2005. (Format: TXT=23694 bytes) (Status: EXPERIMENTAL)

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport



CHAPTER 7

Implementing RSVP for MPLS-TE

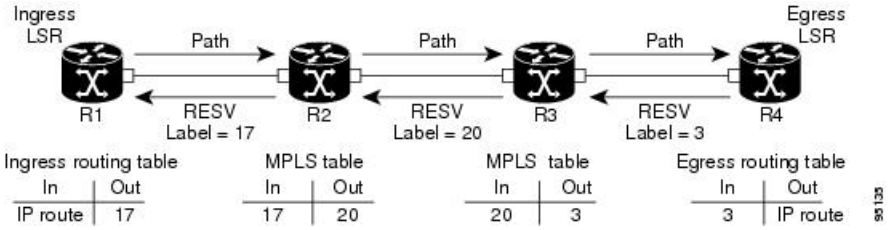
Resource Reservation Protocol (RSVP) is a signaling protocol that enables systems to request resource reservations from the network. RSVP processes protocol messages from other systems, processes resource requests from local clients, and generates protocol messages. As a result, resources are reserved for data flows on behalf of local and remote clients. RSVP creates, maintains, and deletes these resource reservations.

MPLS Traffic Engineering (MPLS-TE) learns the topology and resources available in a network and then maps traffic flows to particular paths based on resource requirements and network resources such as bandwidth. MPLS TE builds a unidirectional tunnel from a source to a destination in the form of a label switched path (LSP), which is then used to forward traffic. MPLS-TE uses RSVP to signal LSPs.

Setting up MPLS LSP Using RSVP

The following figure shows how RSVP sets up an LSP from router R1 through router R4 that can be used for TE in an MPLS environment.

Figure 19: MPLS LSP Using RSVP



The LSP setup is initiated when the LSP head node sends path messages to the tail node. The Path messages reserve resources along the path to each node, and creates path states associated with the session on each node. When the tail node receives a path message, it sends a reservation (RESV) message with a label back to the

previous node. The reservation state in each router is considered as a soft state, which means that periodic PATH and RESV messages must be sent at each hop to maintain the state.

When the reservation message arrives at the previous node, it causes the reserved resources to be locked and forwarding entries are programmed with the MPLS label sent from the tail-end node. A new MPLS label is allocated and sent to the next node upstream. When the reservation message reaches the head node, the label is programmed and the MPLS data starts to flow along the path.

Overview of RSVP for MPLS-TE Features

This section provides an overview of the various features of RSVP for MPLS-TE.

RSVP is automatically enabled on interfaces on which MPLS-TE is configured. For MPLS-TE LSPs with bandwidth, the RSVP bandwidth has to be configured on the interfaces. There is no need to configure RSVP, if all MPLS-TE LSPs have zero bandwidth.

RSVP Graceful restart ensures high availability and allows RSVP TE enabled routers to recover RSVP state information from neighbors after a failure in the network.

RSVP requires that the path and reservation state that are set up during LSP signaling must be refreshed by periodically sending refresh messages. Refresh messages are used to synchronize the state between RSVP neighbors and to recover from lost RSVP messages. RSVP refresh reduction feature includes support for reliable messages which are transmitted rapidly when the messages are lost. Summary refresh messages contain information to refresh multiple states and reduces the number of messages required to refresh states.

RSVP messages can be authenticated to ensure that only trusted neighbors can set up reservations.

For detailed information about RSVP for MPLS-TE features, see the *RSVP for MPLS-TE Features- Details* topic.

Bandwidth Reservation Percentage

The Bandwidth Reservation Percentage allows the RSVP interface bandwidth to be specified as percentages of the link's physical bandwidth.

For more information on configuring RSVP bandwidth, refer the *Implementing MPLS Traffic Engineering* chapter in the *MPLS Configuration Guide for Cisco 8000 Series Routers*. For more information on commands for configuring RSVP bandwidth, refer the *RSVP Infrastructure Commands* chapter in the *MPLS Command Reference for Cisco 8000 Series Routers*.

Caveats for Out-of-Sequence

These caveats are listed for out-of-sequence:

- When RSVP messages traverse multiple interface types with different maximum transmission unit (MTU) values, some messages can become out-of-sequence if they are fragmented.
- Packets with some IP options may be reordered.
- Change in QoS configurations may lead to a transient reorder of packets.
- QoS policies can cause a reorder of packets in a steady state.

Because all out-of-sequence messages are dropped, the sender must retransmit them. Because RSVP state timeouts are generally long, out-of-sequence messages during a transient state do not lead to a state timeout.

Keychain Configuration For RSVP Authentication

Before implementing RSVP authentication, you must configure a keychain first. The name of the keychain must be the same as the one used in the keychain configuration. For more information about configuring keychains, see *System Security Configuration Guide for Cisco 8000 Series Routers*.

Note RSVP authentication supports only keyed-hash message authentication code (HMAC) type algorithms.

Configuring RSVP for MPLS-TE

RSVP requires coordination among several routers, establishing exchange of RSVP messages to set up LSPs. Depending on the requirements, RSVP requires some basic configuration described in the following topics:

Configuring Traffic Engineering Tunnel Bandwidth

To configure traffic engineering tunnel bandwidth, you must first set up TE tunnels and configure the reserved bandwidth per interface (there is no need to configure bandwidth for the data channel or the control channel).

Cisco IOS XR software supports two MPLS DS-TE modes: Prestandard and IETF.

Note For prestandard DS-TE you do not need to configure bandwidth for the data channel or the control channel. There is no other specific RSVP configuration required for this application. When no RSVP bandwidth is specified for a particular interface, you can specify zero bandwidth in the LSP setup if it is configured under RSVP interface configuration mode or MPLS-TE configuration mode.

Confirming DiffServ-TE Bandwidth

In RSVP global and subpools, reservable bandwidths are configured per interface to accommodate TE tunnels on the node. Available bandwidth from all configured bandwidth pools is advertised using IGP. RSVP signals the TE tunnel with appropriate bandwidth pool requirements.

Configuration Example

In this example, the **bandwidth** command sets the total reservable bandwidth, the maximum RSVP bandwidth available for a flow and the sub-pool bandwidth for the HundredGigE 0/0/0/3 interface.

```
Router# configure
Router(config)# rsvp interface HundredGigE0/0/0/3
Router(config-rsvp-if)# bandwidth 1000 mbps 100 mbps sub-pool 150 mbps
Router(config-rsvp-if)# commit
```

Global, Interface, and Neighbor Authentication Modes

You can configure global defaults for all authentication parameters including key, window size, and lifetime. These defaults are inherited when you configure authentication for each neighbor or interface. However, you can also configure these parameters individually on a neighbor or interface basis, in which case the global values (configured or default) are no longer inherited.

Note RSVP uses the following rules when choosing which authentication parameter to use when that parameter is configured at multiple levels (interface, neighbor, or global). RSVP goes from the most specific to least specific; that is, neighbor, interface, and global.

Global keys simplify the configuration and eliminate the chances of a key mismatch when receiving messages from multiple neighbors and multiple interfaces. However, global keys do not provide the best security.

Interface keys are used to secure specific interfaces between two RSVP neighbors. Because many of the RSVP messages are IP routed, there are many scenarios in which using interface keys are not recommended. If all keys on the interfaces are not the same, there is a risk of a key mismatch for the following reasons:

- When the RSVP graceful restart is enabled, RSVP hello messages are sent with a source IP address of the local router ID and a destination IP address of the neighbor router ID. Because multiple routes can exist between the two neighbors, the RSVP hello message can traverse to different interfaces.
- When the RSVP fast reroute (FRR) is active, the RSVP Path and Resv messages can traverse multiple interfaces.
- When Generalized Multiprotocol Label Switching (GMPLS) optical tunnels are configured, RSVP messages are exchanged with router IDs as the source and destination IP addresses. Since multiple control channels can exist between the two neighbors, the RSVP messages can traverse different interfaces.

Neighbor-based keys are particularly useful in a network in which some neighbors support RSVP authentication procedures and others do not. When the neighbor-based keys are configured for a particular neighbor, you are advised to configure all the neighbor's addresses and router IDs for RSVP authentication.

Configuring RSVP Message Authentication Globally

The RSVP authentication feature permits neighbors in an RSVP network to use a secure hash algorithm to authenticate all RSVP signaling messages digitally. The authentication is accomplished on a per-RSVP-hop basis using an RSVP integrity object in the RSVP message. The integrity object includes a key ID, a sequence number for messages, and keyed message digest.

You can globally configure the values of authentication parameters including the key-chain, time interval that RSVP maintains security associations with other trusted RSVP neighbors (life time) and maximum number of RSVP authenticated messages that can be received out of sequence (window size). These defaults are inherited for each neighbor or interface.

Configuration Example

In this example, authentication parameters are configured globally on a router. The authentication parameters including authentication key-chain, lifetime, and window size are configured. A valid key-chain should be configured before performing this task.

```
Router# configure
Router(config)# key chain mpls-keys
Router(config-mpls-keys)# commit
Router(config-mpls-keys)# exit
Router(config)# rsvp authentication
Router(config-rsvp-auth)# key-source key-chain mpls-keys
Router(config-rsvp-auth)# life-time 2000
Router(config-rsvp-auth)# window-size 33
```

Verification

Verify the configuration of authentication parameters using the following command.

```
Router# show rsvp authentication detail

  RSVP Authentication Information:
    Source Address:         3.0.0.1
    Destination Address:    3.0.0.2
    Neighbour Address:      3.0.0.2
    Interface:              HundredGigabitEthernet 0/0/0/3
    Direction:              Send
    LifeTime:               2000 (sec)
    LifeTime left:          1305 (sec)
    KeyType:                Static Global KeyChain
    Key Source:             mpls-keys
    Key Status:             No error
    KeyID:                  1
    Digest:                 HMAC MD5 (16)
    window-size:            33
Challenge:              Not supported
    TX Sequence:            5023969459702858020 (0x45b8b99b00000124)
    Messages successfully authenticated:    245
    Messages failed authentication:         0
```

Configuring RSVP Authentication for an Interface

You can individually configure the values of RSVP authentication parameters including key-chain, life time, and window size on an interface. Interface specific authentication parameters are used to secure specific interfaces between two RSVP neighbors.

Configuration Example

This example configures authentication key-chain, life time for the security association, and window size on an interface. A valid key-chain should be already configured to use it as part of this task.

```
Router# configure
Router(config)# rsvp interface HundredGigE0/0/0/3
Router(config-rsvp-if)# authentication
Router(config-rsvp-if-auth)# key-source key-chain mpls-keys
Router(config-rsvp-if-auth)# life-time 2000
Router(config-rsvp-if-auth)# window-size 33
Router(config-rsvp-if-auth)# commit
```

Cisco IOS XR Release 7.11.1 introduces support to disable RSVP authentication.

Verification

Verify the configuration of authentication parameters using the following command.

```
Router# show rsvp authentication detail
```

```
RSVP Authentication Information:
  Source Address:         3.0.0.1
  Destination Address:    3.0.0.2
  Neighbour Address:      3.0.0.2
  Interface:              HundredGigabitEthernet 0/0/0/3
  Direction:              Send
  LifeTime:               2000 (sec)
  LifeTime left:          1305 (sec)
  KeyType:                Static Global KeyChain
  Key Source:             mpls-keys
  Key Status:             No error
  KeyID:                  1
  Digest:                 HMAC MD5 (16)
  window-size:            33
  Challenge:              Not supported
  TX Sequence:            5023969459702858020 (0x45b8b99b00000124)
  Messages successfully authenticated:    245
  Messages failed authentication:         0
```

Configuring RSVP Authentication on a Neighbor

You can individually configure the values of RSVP authentication parameters including key-chain, life time, and window size on a neighbor.

Configuration Example

This example configures the authentication key-chain, life time for the security association, and window size on a RSVP neighbor. A valid key-chain should be already configured to use it as part of this task.

```
Router# configure
Router(config)# rsvp neighbor 10.0.0.1 authentication
Router(config-rsvp-nbor-auth)# key-source key-chain mpls-keys
Router(config-rsvp-nbor-auth)# life-time 2000
Router(config-rsvp-nbor-auth)# window-size 33
Router(config-rsvp-nbor-auth)# commit
```

Verification

Verify the configuration of authentication parameters using the following command.

```
Router# show rsvp authentication detail
```

```
  RSVP Authentication Information:
    Neighbour Address:      10.0.0.1
    Interface:              HundredGigabitEthernet 0/0/0/3
    Direction:              Send
    LifeTime:               2000 (sec)
    LifeTime left:          1205 (sec)
    KeyType:                Static Global KeyChain
    Key Source:             mpls-keys
    Key Status:             No error
    KeyID:                  1
    Digest:                 HMAC MD5 (16)
    window-size:            33
    Challenge:              Not supported
```

RSVP Authentication by Using All the Modes: Example

The configuration example shows how to perform the following functions:

- Authenticates all RSVP messages.
- Authenticates the RSVP messages to or from 10.0.0.1 by setting the keychain for the **key-source key-chain** command to nbr_keys, SA lifetime is set to 3600, and the default window-size is set to 1.
- Authenticates the RSVP messages not to or from 10.0.0.1 by setting the keychain for the **key-source key-chain** command to default_keys, SA lifetime is set to 3600, and the window-size is set 64 when using GigabitEthernet0/6/0/0; otherwise, the default value of 1 is used.

```
rsvp
 interface GigabitEthernet0/6/0/0
  authentication
   window-size 64
  !
 !
 neighbor 10.0.0.1
  authentication
   key-source key-chain nbr_keys
  !
 !
 authentication
  key-source key-chain default_keys
  life-time 3600
 !
!
```

Note If a keychain does not exist or contain valid keys, this is considered a configuration error because signaling fails. However, this can be intended to prevent signaling. For example, when using the above configuration, if the nbr_keys does not contain valid keys, all signaling with 10.0.0.1 fails.

Configuring Graceful Restart

RSVP graceful restart provides a mechanism to ensure high availability (HA), which allows detection and recovery from failure conditions for systems running Cisco IOS XR software, and ensures non-stop forwarding services. RSVP graceful restart is based on RSVP hello messages and allows RSVP TE enabled routers to recover RSVP state information from neighbors after a failure in the network. RSVP uses a Restart Cap object (RSVP RESTART) in hello messages in which restart and recovery times are specified to advertise the restart capability of a node. The neighboring node helps a restarting node by sending a Recover Label object to recover the forwarding state of the restarting node.

You can configure standard graceful restart which is based on node-id address based hello messages and also interface-based graceful restart which is interface-address based hello messages.

Configuration Example

In this example, RSVP-TE is already enabled on the router nodes on a network and graceful restart needs to be enabled on the router nodes for failure recovery. Graceful restart is configured globally to enabled node-id address based hello messages and also on a router interface to support interface-address based hello messages.

```
Router# configure
Router(config)# rsvp
Router(config-rsvp)# signalling graceful-restart
Router(config-rsvp)# interface HundredGigabitEthernet 0/0/0/3
```

```
Router(config-rsvp-if)# signalling hello graceful-restart interface-based
Router(config-rsvp-if)# commit
```

Verification

Use the following commands to verify that graceful restart is enabled.

```
Router# show rsvp graceful-restart

 Graceful restart: enabled Number of global neighbors: 1
 Local MPLS router id: 192.168.55.55
 Restart time: 60 seconds  Recovery time: 120 seconds
 Recovery timer: Not running
 Hello interval: 5000 milliseconds Maximum Hello miss-count: 4
```

```
Router# show rsvp graceful-restart neighbors detail

 Neighbor: 192.168.77.77 Source: 192.168.55.55 (MPLS)
   Hello instance for application MPLS
     Hello State: UP        (for 00:20:52)
     Number of times communications with neighbor lost: 0
     Reason: N/A
Recovery State: DONE
   Number of Interface neighbors: 1
     address: 192.168.55.0
   Restart time: 120 seconds  Recovery time: 120 seconds
   Restart timer: Not running
   Recovery timer: Not running
   Hello interval: 5000 milliseconds  Maximum allowed missed Hello messages: 4
```

Change the Restart-Time: Example

The example shows how to change the restart time that is advertised in hello messages sent to neighbor nodes.

```
rsvp signalling graceful-restart restart-time 200
```

Configuring Refresh Reduction

RSVP Refresh Reduction improves the reliability of Resource Reservation Protocol (RSVP) signaling to enhance network performance and message delivery and it is enabled by default. Refresh reduction is used with a neighbor only if the neighbor supports it. You can also disable refresh reduction on an interface if you want.

This feature ensures reliable delivery of RSVP messages when network traffic is disrupted. To ensure that its message is delivered to its neighbor, RSVP requests the neighbor to send an acknowledgment message by a given time duration. If it doesn't receive the acknowledgment, it resends the message and doubles its current wait time. After 5 attempts, RSVP stops retransmitting the message to the neighbor.

Configuration Example

The example shows how to configure the various parameters available for the refresh reduction feature.

The following parameters are configured to change their default values:

- refresh interval
- number of refresh messages a node can miss

- retransmit time
- acknowledgment hold time
- acknowledgment message size
- refresh message summary size

```
Router# configure
Router(config)# rsvp
Router(config-rsvp)# interface HundredGigabitEthernet 0/0/0/3
Router(config-rsvp-if)# signalling refresh interval 40
Router(config-rsvp-if)# signalling refresh missed 6
Router(config-rsvp-if)# signalling refresh reduction reliable retransmit-time 2000
Router(config-rsvp-if)# signalling refresh reduction reliable ack-hold-time 1000
Router(config-rsvp-if)# signalling refresh reduction reliable ack-max-size 1000
Router(config-rsvp-if)# signalling refresh reduction summary  max-size 1500
Router(config-rsvp-if)# commit
```

Change the Hello Interval: Example

The example shows how to change the interval at which RSVP graceful restart hello messages are sent per neighbor, and change the number of hellos missed before the neighbor is declared down.

```
rsvp signalling hello graceful-restart refresh interval 4000
rsvp signalling hello graceful-restart refresh misses 4
```

Disable Refresh Reduction: Example

If the peer node does not support refresh reduction, or for any other reason you want to disable refresh reduction on an interface, the example shows how to disable refresh reduction on that interface.

```
Router(config)# rsvp
Router(config-rsvp)# interface hundredGigE 0/0/0/0
Router(config-rsvp-if)# signalling refresh reduction disable
```

RSVP Prefix Filtering

Two procedures are provided to show how RSVP Prefix Filtering is associated:

Configuring ACL Based Prefix Filtering

You can configure extended access lists (ACLs) to forward, drop, or perform normal processing on RSVP router-alert (RA) packets. For each incoming RSVP RA packet, RSVP inspects the IP header and attempts to match the source or destination IP addresses with a prefix configured in an extended ACL. If there is no explicit permit or explicit deny, the ACL infrastructure returns an implicit deny by default. By default, RSVP processes the packet if the ACL match yields an implicit (default) deny.

Configuration Example

This example configures ACL based prefix filtering on RSVP RA packets. When RSVP receives a RA packet from source address 10.0.0.1 it is forwarded and packets destined to the IP address 172.16.0.1 are dropped.

```
Router# configure
Router(config)# ipv4 access-list rsvpac1
Router(config-ipv4-acl)# 10 permit ip host 10.0.0.1 any
Router(config-ipv4-acl)# 20 deny ip any host 172.16.0.1

Router# configure
Router(config)# rsvp
Router(config-rsvp)# signalling prefix-filtering access-list rsvp-acl
Router(config-rsvp)# commit
```

Verification

Verify the configuration of ACL based prefix filtering

```
Router# show rsvp counters prefix-filtering access-list rsvp-ac1

  ACL:rsvp-ac1            Forward          Local          Drop           Total
  Path                      0              0            0              0
  PathTear                  0              0            0              0
  ResvConfirm               0              0            0              0
  Total                     0              0            0
```

Configuring RSVP Packet Dropping

You can configure extended access lists (ACLs) to forward, drop, or perform normal processing on RSVP router-alert (RA) packets. By default, RSVP processes the RA packets even if the ACL match yields an implicit deny. You can configure RSVP to drop RA packets when the ACL matches results in an implicit deny.

Configuration Example

This example configures ACL based prefix filtering on RSVP RA packets. When RSVP receives a RA packet from source address 10.0.0.1 it is forwarded and packets destined to the IP address 172.16.0.1 are dropped. RA packets are dropped if the ACL matches results in an implicit deny.

```
Router# configure
Router(config)# ipv4 access-list rsvpac1
Router(config-ipv4-acl)# 10 permit ip host 10.0.0.1 any
Router(config-ipv4-acl)# 20 deny ip any host 172.16.0.1
Router(config-ipv4-acl)# exit
Router(config)# rsvp
Router(config-rsvp)# signalling prefix-filtering default-deny-action drop
Router(config-rsvp)# commit
```

Verification

Verify the configuration of RSVP packet drop using the following command.

```
Router# show rsvp counters prefix-filtering access-list rsvpac1

  ACL: rsvpac1            Forward          Local          Drop           Total
  Path                      4              1            0              5
  PathTear                  0              0            0              0
  ResvConfirm               0              0            0              0
  Total                     4              1            0              5
```

Enabling RSVP Traps

By implementing the RSVP MIB, you can use SNMP to access objects belonging to RSVP. You can also specify two traps (NewFlow and LostFlow) which are triggered when a new flow is created or deleted. RSVP MIBs are automatically enabled when you turn on RSVP, but you need to enable RSVP traps.

Configuration Example

This example shows how to enable RSVP MIB traps when a flow is deleted or created and also how to enable both the traps.

```
Router# configure
Router(config)# snmp-server traps rsvp lost-flow
Router(config)# snmp-server traps rsvp new-flow
Router(config)# snmp-server traps rsvp all
Router(config)# commit
```

Eliminating Security Associations for RSVP Authentication

To eliminate RSVP authentication SA's, use the **clear rsvp authentication** command. To eliminate RSVP counters for each SA, use the **clear rsvp counters authentication** command.

RSVP for MPLS-TE Features - Details

RSVP Graceful Restart Operation

RSVP graceful restart is based on RSVP hello messages. Hello messages are exchanged between the router and its neighbor nodes. Each neighbor node can autonomously issue a hello message containing a hello request object. A receiver that supports the hello extension replies with a hello message containing a hello acknowledgment (ACK) object. If the sending node supports state recovery, a Restart Cap object that indicates a node's restart capability is also carried in the hello messages. In the Restart Cap object, the restart time and the recovery time is specified. The restart time is the time after a loss in Hello messages within which RSVP hello session can be re-established. The recovery time is the time that the sender waits for the recipient to re-synchronize states after the re-establishment of hello messages.

For graceful restart, the hello messages are sent with an IP Time to Live (TTL) of 64. This is because the destination of the hello messages can be multiple hops away. If graceful restart is enabled, hello messages (containing the restart cap object) are send to an RSVP neighbor when RSVP states are shared with that neighbor. If restart cap objects are sent to an RSVP neighbor and the neighbor replies with hello messages containing the restart cap object, the neighbor is considered to be graceful restart capable. If the neighbor does not reply with hello messages or replies with hello messages that do not contain the restart cap object, RSVP backs off sending hellos to that neighbor. If a hello Request message is received from an unknown neighbor, no hello ACK is sent back.

RSVP Authentication

Network administrators need the ability to establish a security domain to control the set of systems that initiates RSVP requests. The RSVP authentication feature permits neighbors in an RSVP network to use a secure hash to sign all RSVP signaling messages digitally, thus allowing the receiver of an RSVP message to verify the sender of the message without relying solely on the sender's IP address.

The signature is accomplished on a per-RSVP-hop basis with an RSVP integrity object in the RSVP message as defined in RFC 2747. The integrity object includes a key ID, a sequence number for messages, and keyed message digest. This method provides protection against forgery or message modification. However, the receiver must know the security key used by the sender to validate the digital signature in the received RSVP message. Network administrators manually configure a common key for each RSVP neighbor on the shared network. The sending and receiving systems maintain a security association for each authentication key that they share. For detailed information about different security association parameters, see the **Security Association Parameters** table.

You can configure global defaults for all authentication parameters including key, window size, and lifetime. These defaults are inherited when you configure authentication for each neighbor or interface. However, you can also configure these parameters individually on a neighbor or interface basis, in which case the global values (configured or default) are no longer inherited.

Interface and neighbor interface modes unless explicitly configured, inherit the parameters from global configuration mode as follows:

- Window-size is set to 1.
- Lifetime is set to 1800.
- key-source key-chain command is set to none or disabled.

The following situations explain how to choose between global, interface, or neighbor configuration modes:

- Global configuration mode is optimal when a router belongs to a single security domain (for example, part of a set of provider core routers). A single common key set is expected to be used to authenticate all RSVP messages.
- Interface, or neighbor configuration mode, is optimal when a router belongs to more than one security domain. For example, a provider router is adjacent to the provider edge (PE), or a PE is adjacent to an edge device. Different keys can be used but not shared.

A security association (SA) is a collection of information that is required to maintain secure communications with a peer. The following table lists the main parameters that defines a security association

Table 12: Security Association Parameters

Security Association Parameter	Description
src	IP address of the sender.
dst	IP address of the final destination.
interface	Interface of the security association.
direction	Send or receive type of the security association.
Lifetime	Expiration timer value that is used to collect unused security association data.
Sequence Number	Last sequence number that was either sent or accepted (dependent of the direction type).
key-source	Source of keys for the configurable parameter.

Security Association Parameter	Description
keyID	Key number (returned form the key-source) that was last used.
Window Size	Specifies the maximum number of authenticated messages that can be received out of order.
Window	Specifies the last *window size* value sequence number that is received or accepted.

MPLS-TE LSP OOR

The MPLS-TE LSP OOR function adds capability for the RSVP-TE control plane to track the LSP scale of transit routers, so that it can take a specific set of (pre-configured) actions when threshold limits are crossed, and inform other routers in the network. MPLS-TE keeps track of the number of transit LSPs set up through the router. The limits do not apply to ingress and egress LSP routers since they are driven by explicit configuration. In other words, the configuration determines how many egress or ingress LSPs a router has. For midpoint routers, the number is a function of the topology, the links metrics, and links' bandwidth.

State Transition Triggers - The LSP OOR state transition is triggered by checking the total transit LSP count and the unprotected count. If either count crosses the threshold, the state transition is triggered. If both counts cross the limit, the more critical state is chosen. Each limit will have a value for the *Yellow* threshold and a value for the *Red* threshold. When these thresholds are crossed, the configured MPLS-TE LSP OOR actions take effect. Similarly, the transition to *Green* state occurs when the LSP numbers drop.

LSP OOR State Dampening - The reason for LSP OOR State Dampening is that the number of accepted LSPs would be at the threshold and once an LSP is deleted, the state goes back from Red to Yellow, and a new LSP is setup and the state goes back to Red.

The solution is to introduce dampening when there is a state transition from Red to Yellow or from Yellow to Green. Whenever the transit number of LSPs crosses down a threshold, a timer is started for 10 seconds. After the timer expires, the new state is computed and moved to it. The timer is stopped if the transit number threshold is crossed (up) again. The transition from a state to a more severe state is not dampened.

Low and High Priority LSPs - When the LSP OOR is in yellow or red state, new high priority LSPs will not preempt low priority LSPs. Preemption can still occur but only for bandwidth reasons. In other words, if the router is in Red state where one of the actions is to reject any new LSP, the new high-priority LSPs are rejected even if there is an established low-priority LSP. The low-priority LSP is not removed to make room for the high-priority one.

Configuration Limit - Setting the configured limit to a value that is smaller than the current number of LSPs will trigger state transition but will not cause existing LSPs to be deleted or preempted. Setting the configured limit to a value that is larger than the current number of LSPs takes the node out of LSP OOR state. When an LSP cannot be admitted due to LSP OOR, the LSRs send Path Error messages to the LERs.

Event Logging - This is generated when the system transitions across OOR states, such as a resource change into an *yellow* or *red* state. Reporting level for *Red* is critical (1), and for yellow is warning (4). The following example shows that the count has crossed the threshold of 5000.

```
RP/0/RP1/CPU0:May 15 17:05:48 PDT: te_control[1034]: %ROUTING-MPLS_TE-4-LSP_OOR :

Transit LSP resources changed to Yellow.
Total transit: configured threshold 5000; actual count 5001;
Unprotected transit: configured threshold 4294967295; actual count 0
```

When the resource comes out of OOR, it will report as *green*.

Configuration Example

```
mpls traffic-eng
 lsp-oor
  green
   action accept reopt-lsp
   action flood available-bw 20
   recovery-duration
   action admit lsp-min-bw   X   -- > (in kbps, a lower limit than yellow and red state)

  yellow
   transit-all threshold 75000
   action accept reopt-lsp
   action flood available-bw 0
   action admit lsp-min-bw  Y

  red
   transit-all threshold 90000
   action flood available-bw 0
   action admit lsp-min-bw Z
```

The LSP OOR threshold values are set to yellow as 75000 and red as 90000. When these thresholds are crossed, corresponding actions are applied to all the TE interfaces.

Note The default values of the above thresholds are infinite.

When the LSP OOR *yellow* state is reached, the **accept reopt-lsp** action, **flood available-bw 0** action and **admit lsp-min-bw** actions are activated. This allows headend routers to reoptimize existing LSPs through, but doesn't allow new LSPs to get established. Also, MPLS-TE advertises zero bandwidth out of all interfaces, making this transit router less preferable for new LSPs. To handle a sudden burst of new LSPs that get signaled, the **action admit lsp-min-bw** function ensures only a small number of high bandwidth LSPs get provisioned through the affected router. When the red threshold state is crossed, the **flood available-bw 0** and **admit lsp-min-bw** actions prevent any additional or reoptimized transit LSPs from getting set up through the affected router.

Additional References

For additional information related to RSVP, refer to the following references:

Related Documents

Related Topic	Document Title
RSVP Infrastructure Commands	*RSVP Infrastructure Commands* module in *MPLS Command Reference for Cisco 8000 Series Routers*.
MPLS Traffic Engineering Commands	*MPLS Traffic Engineering commands* module in *MPLS Command Reference for Cisco 8000 Series Routers*.

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport