MPLS Basic Configuration Guide, Cisco IOS XE 16 (Cisco ASR 900 Series)

First Published: 2017-05-04

Last Modified: 2019-08-01

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

CHAPTER 1

Multiprotocol Label Switching (MPLS) on Cisco Routers

This document describes commands for configuring and monitoring Multiprotocol Label Switching (MPLS) functionality on Cisco routers and switches. This document is a companion to other feature modules describing other MPLS applications.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About MPLS

MPLS Overview

Multiprotocol label switching (MPLS) combines the performance and capabilities of Layer 2 (data link layer) switching with the proven scalability of Layer 3 (network layer) routing. MPLS enables service providers to meet the challenges of explosive growth in network utilization while providing the opportunity to differentiate services without sacrificing the existing network infrastructure. The MPLS architecture is flexible and can be employed in any combination of Layer 2 technologies. MPLS support is offered for all Layer 3 protocols, and scaling is possible well beyond that typically offered in today's networks.

MPLS efficiently enables the delivery of IP services over an ATM switched network. MPLS supports the creation of different routes between a source and a destination on a purely router-based Internet backbone. By incorporating MPLS into their network architecture, service providers can save money, increase revenue and productivity, provide differentiated services, and gain competitive advantages.

Functional Description of MPLS

Label switching is a high-performance packet forwarding technology that integrates the performance and traffic management capabilities of data link layer (Layer 2) switching with the scalability, flexibility, and performance of network layer (Layer 3) routing.

Label Switching Functions

In conventional Layer 3 forwarding mechanisms, as a packet traverses the network, each router extracts all the information relevant to forwarding the packet from the Layer 3 header. This information is then used as an index for a routing table lookup to determine the next hop for the packet.

In the most common case, the only relevant field in the header is the destination address field, but in some cases, other header fields might also be relevant. As a result, the header analysis must be done independently at each router through which the packet passes. In addition, a complicated table lookup must also be done at each router.

In label switching, the analysis of the Layer 3 header is done only once. The Layer 3 header is then mapped into a fixed length, unstructured value called a *label* .

Many different headers can map to the same label, as long as those headers always result in the same choice of next hop. In effect, a label represents a *forwarding equivalence class* --that is, a set of packets which, however different they may be, are indistinguishable by the forwarding function.

The initial choice of a label need not be based exclusively on the contents of the Layer 3 packet header; for example, forwarding decisions at subsequent hops can also be based on routing policy.

Once a label is assigned, a short label header is added at the front of the Layer 3 packet. This header is carried across the network as part of the packet. At subsequent hops through each MPLS router in the network, labels are swapped and forwarding decisions are made by means of MPLS forwarding table lookup for the label carried in the packet header. Hence, the packet header does not need to be reevaluated during packet transit through the network. Because the label is of fixed length and unstructured, the MPLS forwarding table lookup process is both straightforward and fast.

Distribution of Label Bindings

Each label switching router (LSR) in the network makes an independent, local decision to determine a label value to represent a forwarding equivalence class. This association is known as a label binding. Each LSR informs its neighbors of the label bindings it has made.

When a labeled packet is being sent from LSR A to the neighboring LSR B, the label value carried by the IP packet is the label value that LSR B assigned to represent the forwarding equivalence class of the packet. Thus, the label value changes as the IP packet traverses the network.

The awareness of label bindings by neighbouring routers is facilitated using the following protocols:

- Label Distribution Protocol (LDP) - Enables peer LSRs in an MPLS network to exchange label binding information for supporting hop-by-hop forwarding in an MPLS network.

- Tag Distribution Protocol (TDP) - Supports MPLS forwarding along normally routed paths.
- Resource Reservation Protocol (RSVP) - Supports MPLS traffic engineering.
- Border Gateway Protocol (BGP) - Supports MPLS virtual private networks (VPNs) .

Benefits of MPLS

MPLS provides the following major benefits to service provider networks:

Scalable support for Virtual Private Networks (VPNs)--MPLS enables VPN services to be supported in service provider networks, thereby greatly accelerating Internet growth.

The use of MPLS for VPNs provides an attractive alternative to the building of VPNs by means of either ATM or Frame Relay permanent virtual circuits (PVCs) or various forms of tunneling to interconnect routers at customer sites.

Unlike the PVC VPN model, the MPLS VPN model is highly scalable and can accommodate increasing numbers of sites and customers. The MPLS VPN model also supports “any-to-any” communication among VPN sites without requiring a full mesh of PVCs or the backhauling (suboptimal routing) of traffic across the service provider network. For each MPLS VPN user, the service provider’s network appears to function as a private IP backbone over which the user can reach other sites within the VPN organization, but not the sites of any other VPN organization.

From a user perspective, the MPLS VPN model enables network routing to be dramatically simplified. For example, rather than having to manage routing over a topologically complex virtual backbone composed of many PVCs, an MPLS VPN user can generally employ the service provider’s backbone as the default route in communicating with all of the other VPN sites.

Explicit routing capabilities (also called constraint-based routing or traffic engineering)--Explicit routing employs “constraint-based routing,” in which the path for a traffic flow is the shortest path that meets the resource requirements (constraints) of the traffic flow.

In MPLS traffic engineering, factors such as bandwidth requirements, media requirements, and the priority of one traffic flow versus another can be taken into account. These traffic engineering capabilities enable the administrator of a service provider network to

- Control traffic flow in the network
- Reduce congestion in the network
- Make best use of network resources

Thus, the network administrator can specify the amount of traffic expected to flow between various points in the network (thereby establishing a traffic matrix), while relying on the routing system to

- Calculate the best paths for network traffic
- Set up the explicit paths to carry the traffic

Support for IP routing on ATM switches (also called IP and ATM integration)--MPLS enables an ATM switch to perform virtually all of the functions of an IP router. This capability of an ATM switch stems from the fact that the MPLS forwarding paradigm, namely, label swapping, is exactly the same as the forwarding paradigm provided by ATM switch hardware.

The key difference between a conventional ATM switch and an ATM label switch is the control software used by the latter to establish its virtual channel identifier (VCI) table entries. An ATM label switch uses IP routing protocols and the Tag Distribution Protocol (TDP) to establish VCI table entries.

An ATM label switch can function as a conventional ATM switch. In this dual mode, the ATM switch resources (such as VCI space and bandwidth) are partitioned between the MPLS control plane and the ATM control plane. The MPLS control plane provides IP-based services, while the ATM control plane supports ATM-oriented functions, such as circuit emulation or PVC services.

Restrictions for MPLS

Cisco ASR 900 Routers with RSP3 module can push a maximum of 4 MPLS labels in the egress direction. This includes service labels (L3VPN, L2VPN, 6PE, 6VPE), RFC 3107 BGP-LU label, and IGP labels for FRR primary or backup paths. However, Flow Aware Transport (FAT) pseudowire MPLS label is not included in this 4 label limit. FAT pseudowire MPLS label if configured, can be pushed in addition to the 4 labels mentioned.

How to Configure MPLS

This section explains how to perform the basic configuration required to prepare a router for MPLS switching and forwarding.

Configuration tasks for other MPLS applications are described in the feature module documentation for the application.

Configuring a Router for MPLS Switching

MPLS switching on Cisco routers requires that Cisco Express Forwarding be enabled.

For more information about Cisco Express Forwarding commands, see the Cisco IOS Switching Command Reference.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef distributed**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ip cef distributed** **Example:** `Device(config)# ip cef distributed`	Enables Cisco Express Forwarding on the route processor card.

Verifying Configuration of MPLS Switching

To verify that Cisco Express Forwarding has been configured properly, issue the **show ip cef summary** command, which generates output similar to that shown below:

SUMMARY STEPS

1. **show ip cef summary**

DETAILED STEPS

show ip cef summary

Example:

```
Router# show ip cef summary
IP CEF with switching (Table Version 49), flags=0x0
  43 routes, 0 resolve, 0 unresolved (0 old, 0 new)
  43 leaves, 49 nodes, 56756 bytes, 45 inserts, 2 invalidations
  2 load sharing elements, 672 bytes, 2 references
  1 CEF resets, 4 revisions of existing leaves
  4 in-place modifications
  refcounts:  7241 leaf, 7218 node
Adjacency Table has 18 adjacencies
Router#
```

Configuring a Router for MPLS Forwarding

MPLS forwarding on Cisco routers requires that forwarding of IPv4 packets be enabled.

For more information about MPLS forwarding commands, see the *Multiprotocol Label Switching Command Reference*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/subslot /port* [**.** *subinterface*]
4. **mpls ip**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot*/*subslot* /*port* [**.** *subinterface*] **Example:** `Device(config)# interface gigabitethernet 4/0/0`	Specifies the Gigabit Ethernet interface and enters interface configuration mode.
Step 4	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables MPLS forwarding of IPv4 packets along normally routed paths for the Gigabit Ethernet interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

What to do next

Configure either of the following:

- MPLS Label Distribution Protocol (LDP). For information about configuring MPLS LDP, see the *MPLS Label Distribution Protocol Configuration Guid*e.
- Static labels. For information about configuring static labels, see *MPLS Static Labels*.

Verifying Configuration of MPLS Forwarding

To verify that MPLS forwarding has been configured properly, issue the **show mpls interfaces detail** command, which generates output similar to that shown below:

SUMMARY STEPS

1. **show mpls interfaces detail**

DETAILED STEPS

show mpls interfaces detail

Example:

```
Device# show mpls interfaces detail

Interface GigabitEthernet1/0/0:
         IP labeling enabled (ldp)
         LSP Tunnel labeling not enabled
         MPLS operational
         MTU = 1500
Interface POS2/0/0:
         IP labeling enabled (ldp)
         LSP Tunnel labeling not enabled
         MPLS not operational
         MTU = 4470
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference*

Standards

Standard	Title
The supported standards applicable to the MPLS applications appear in the respective feature module for the application.	--

MIBs

MIB	MIBs Link
The supported MIBs applicable to the MPLS applications appear in the respective feature module for the application.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
The supported RFCs applicable to the MPLS applications appear in the respective feature module for the application.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	*Support & Downloads*

Feature Information for MPLS on Cisco Routers

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for MPLS on Cisco Routers

Feature Name	Releases	Feature Information
MPLS (Multiprotocol Label Switching)	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.5S	Multiprotocol label switching (MPLS) combines the performance and capabilities of Layer 2 (data link layer) switching with the proven scalability of Layer 3 (network layer) routing. MPLS enables service providers to meet the challenges of explosive growth in network utilization while providing the opportunity to differentiate services without sacrificing the existing network infrastructure. In Cisco IOS XE Release 2.1, this feature was introduced. In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router. The following commands were introduced or modified: **interface atm**, **mpls atm control-vc**, **mpls atm vpi**, **mpls ip** (global configuration), **mpls ip** (interface configuration), **mpls ip default-route**, **mpls ip propagate-ttl**, **mpls ip ttl-expiration pop**, **mpls label range**, **mpls mtu**, **show mpls forwarding-table**, **show mpls interfaces**, **show mpls label range**, **debug mpls adjacency**, **debug mpls events**, **debug mpls lfib cef**, **debug mpls lfib enc**, **debug mpls lfib lsp**, **debug mpls lfib state**, **debug mpls lfib struct**, **debug mpls packets**.

Glossary

BGP --Border Gateway Protocol. The predominant interdomain routing protocol used in IP networks.

Border Gateway Protocol --See BGP.

FIB --Forwarding Information Base. A table that contains a copy of the forwarding information in the IP routing table.

Forwarding Information Base --See FIB.

label --A short, fixed-length identifier that tells switching nodes how the data (packets or cells) should be forwarded.

label binding --An association between a label and a set of packets, which can be advertised to neighbors so that a label switched path can be established.

Label Distribution Protocol --See LDP.

Label Forwarding Information Base --See LFIB.

label imposition --The act of putting the first label on a packet.

label switching router --See LSR.

LDP --Label Distribution Protocol. The protocol that supports MPLS hop-by-hop forwarding by distributing bindings between labels and network prefixes.

LFIB --Label Forwarding Information Base. A data structure in which destinations and incoming labels are associated with outgoing interfaces and labels.

LSR --label switching router. A Layer 3 router that forwards a packet based on the value of an identifier encapsulated in the packet.

MPLS --Multiprotocol Label Switching. An industry standard on which label switching is based.

MPLS hop-by-hop forwarding --The forwarding of packets along normally routed paths using MPLS forwarding mechanisms.

Multiprotocol Label Switching --See MPLS.

Resource Reservation Protocol --See RSVP.

RIB --Routing Information Base. A common database containing all the routing protocols running on a router.

Routing Information Base --See RIB.

RSVP --Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

traffic engineering --Techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods were used.

Virtual Private Network --See VPN.

VPN --Virtual Private Network. A network that enables IP traffic to use tunneling to travel securely over a public TCP/IP network.

CHAPTER 2

MPLS Transport Profile

Note This chapter is not applicable on the ASR 900 RSP3 Module.

Multiprotocol Label Switching (MPLS) Transport Profile (TP) enables you to create tunnels that provide the transport network service layer over which IP and MPLS traffic traverses. MPLS-TP tunnels enable a transition from Synchronous Optical Networking (SONET) and Synchronous Digital Hierarchy (SDH) time-division multiplexing (TDM) technologies to packet switching to support services with high bandwidth requirements, such as video.

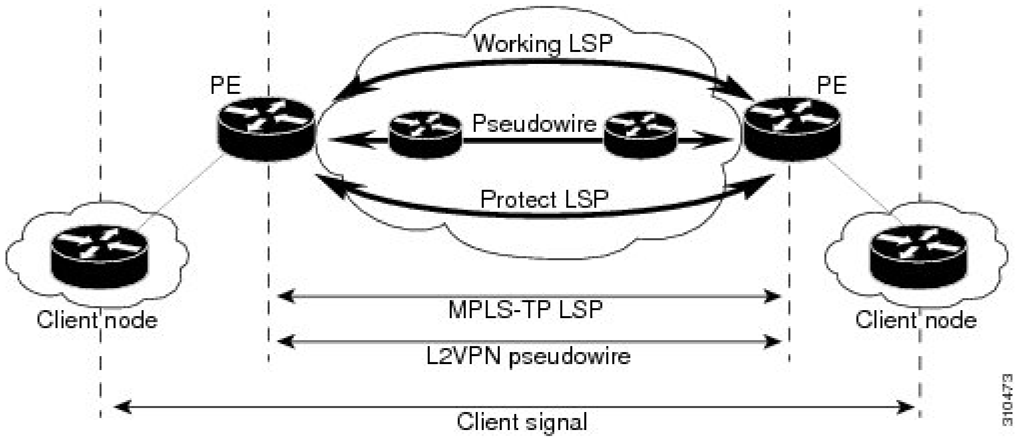
Restrictions for MPLS-TP on the Cisco ASR 900 Series Routers

- Multiprotocol Label Switching Transport Profile (MPLS-TP) penultimate hop popping is *not* supported. Only ultimate hop popping is supported, because label mappings are configured at the MPLS-TP endpoints
- IPv6 addressing is *not* supported.
- VCCV BFD is *not* supported.
- Layer 2 Virtual Private Network (L2VPN) interworking is *not* supported.
- Local switching with Any Transport over MPLS (AToM) pseudowire as a backup is *not* supported.
- L2VPN pseudowire redundancy to an AToM pseudowire by one or more attachment circuits is *not* supported.
- Pseudowire ID Forward Equivalence Class (FEC) type 128 is supported, but generalized ID FEC type 129 is *not* supported
- Maximum virtual circuits (VC) supported for MPLS-TP is 2000.

Information About MPLS-TP

How MPLS Transport Profile Works

Multiprotocol Label Switching Transport Profile (MPLS-TP) tunnels provide the transport network service layer over which IP and MPLS traffic traverses. MPLS-TP tunnels help transition from Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) and Time Division Multiplexing (TDM) technologies to packet switching to support services with high bandwidth utilization and lower cost. Transport networks are connection-oriented, statically provisioned, and have long-lived connections. Transport networks usually avoid control protocols that change identifiers (like labels). MPLS-TP tunnels provide this functionality through statically provisioned bidirectional label switched paths (LSPs), as shown in the figure below.



MPLS-TP is supported on ATM and TDM pseudowires on the Cisco ASR 903 router. For information, see Configuring the Pseudowire Class.

MPLS-TP Path Protection

MPLS-TP label switched paths (LSPs) support 1-to-1 path protection. There are two types of LSPs: protect LSPs and working LSPs. You can configure the both types of LSPs when configuring the MPLS-TP tunnel. The working LSP is the primary LSP used to route traffic. The protect LSP acts as a backup for a working LSP. If the working LSP fails, traffic is switched to the protect LSP until the working LSP is restored, at which time forwarding reverts back to the working LSP.

Bidirectional LSPs

Multiprotocol Label Switching Transport Profile (MPLS-TP) label switched paths (LSPs) are bidirectional and co-routed. They comprise of two unidirectional LSPs that are supported by the MPLS forwarding infrastructure. A TP tunnel consists of a pair of unidirectional tunnels that provide a bidirectional LSP. Each unidirectional tunnel can be optionally protected with a protect LSP that activates automatically upon failure conditions.

MPLS Transport Profile Static and Dynamic Multisegment Pseudowires

Multiprotocol Label Switching Transport Profile (MPLS-TP) supports the following combinations of static and dynamic multisegment pseudowires:

- Dynamic-static
- Static-dynamic
- Static-static

MPLS-TP OAM Status for Static and Dynamic Multisegment Pseudowires

With static pseudowires, status notifications can be provided by BFD over VCCV or by the static pseudowire OAM protocol. However, BFD over VCCV sends only attachment circuit status code notifications. Hop-by-hop notifications of other pseudowire status codes are not supported. Therefore, the static pseudowire OAM protocol is preferred

MPLS Transport Profile Links and Physical Interfaces

Multiprotocol Label Switching Transport Profile (MPLS-TP) link numbers may be assigned to physical interfaces only. Bundled interfaces and virtual interfaces are not supported for MPLS-TP link numbers.

The MPLS-TP link creates a layer of indirection between the MPLS-TP tunnel and midpoint LSP configuration and the physical interface. The **mplstp link** command is used to associate an MPLS-TP link number with a physical interface and next-hop node. The MPLS-TP out-links can be configured only on the ethernet interfaces, with either the next hop IPv4 address or next hop mac-address specified.

Multiple tunnels and LSPs may then refer to the MPLS-TP link to indicate that they are traversing that interface. You can move the MPLS-TP link from one interface to another without reconfiguring all the MPLS-TP tunnels and LSPs that refer to the link.

Link numbers must be unique on the router or node.

Tunnel Midpoints

Tunnel LSPs, whether endpoint or midpoint, use the same identifying information. However, it is entered differently.

- At the midpoint, all information for the LSP is specified with the **mpls tp lsp** command for configuring forward and reverse information for forwarding.
- At the midpoint, determining which end is source and which is destination is arbitrary. That is, if you are configuring a tunnel between your device and a coworker’s device, then your device is the source. However, your coworker considers his or her device to be the source. At the midpoint, either device could be considered the source. At the midpoint, the forward direction is from source to destination, and the reverse direction is from destination to source.
- At the endpoint, the local information (source) either comes from the global device ID and global ID, or from the locally configured information using the **tp source** command.
- At the endpoint, the remote information (destination) is configured using the **tp destination** command after you enter the **interface tunnel-tp** *number* command. The **tp destination** command includes the

destination node ID, and optionally the global ID and the destination tunnel number. If you do not specify the destination tunnel number, the source tunnel number is used.

- At the endpoint, the LSP number is configured in working-lsp or protect-lsp submode. The default is 0 for the working LSP and 1 for the protect LSP.
- When configuring LSPs at midpoint devices, ensure that the configuration does not deflect traffic back to the originating node.

MPLS-TP Linear Protection with PSC Support

MPLS-TP Linear Protection with PSC Support Overview

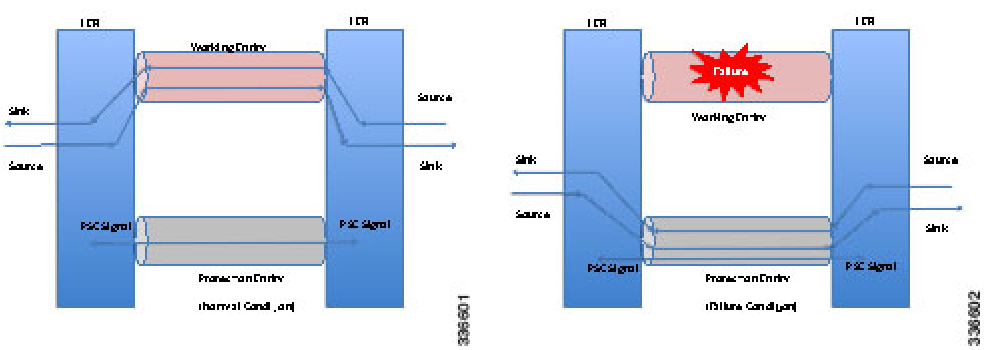
The Multiprotocol Label Switching (MPLS) Transport Profile (TP) enables you to create tunnels that provide the transport network service layer over which IP and MPLS traffic traverse.

Network survivability is the ability of a network to recover traffic deliver following failure, or degradation, of network resources. The MPLS-TP Survivability Framework (RFC-6372) describes the framework for survivability in MPLS-TP networks, focusing on mechanisms for recovering MPLS-TP label switched paths (LSPs)

Linear protection provides rapid and simple protection switching because it can operate between any pair of points within a network. Protection switching is a fully allocated survivability mechanism, meaning that the route and resources of the protection path are reserved for a selected working path or set of working paths. For a point-to-point LSPs, the protected domain is defined as two label edge routers (LERs) and the transport paths that connect them.

Protection switching in a point-to-point domain can be applied to a 1+1, 1:1, or 1:n unidirectional or bidirectional protection architecture. When used for bidirectional switching, the protection architecture must also support a Protection State Coordination (PSC) protocol. This protocol is used to help coordinate both ends of the protected domain in selecting the proper traffic flow. For example, if either endpoint detects a failure on the working transport entity, the endpoint sends a PSC message to inform the peer endpoint of the state condition. The PSC protocol decides what local action, if any, should be taken.

The following figure shows the MPLS-TP linear protection model used and the associated PSC signaling channel for state coordination.



In 1:1 bidirectional protection switching, for each direction, the source endpoint sends traffic on either a working transport entity or a protected transport entity, referred to as a data-path. If the either endpoint detects a failure on the working transport entity, that endpoint switches to send and receive traffic from the protected transport entity. Each endpoint also sends a PSC message to inform the peer endpoint of the state condition.

The PSC mechanism is necessary to coordinate the two transport entity endpoints and implement 1:1 bidirectional protection switching even for a unidirectional failure. The switching of the transport path from working path to protected path can happen because of various failure conditions (such as link down indication (LDI), remote defect indication (RDI), and link failures) or because administrator/operator intervention (such as shutdown, lockout of working/forced switch (FS), and lockout of protection).

Each endpoint LER implements a PSC architecture that consists of multiple functional blocks. They are:

- **Local Trigger Logic:** This receives inputs from bidirectional forwarding detection (BFD), operator commands, fault operation, administration, and maintenance (OAM) and a wait-to-restore (WTR) timer. It runs a priority logic to decide on the highest priority trigger.
- **PSC FSM:** The highest priority trigger event drives the PSC finite state machine (FSM) logic to decide what local action, if any, should be taken. These actions may include triggering path protection at the local endpoint or may simply ignore the event.
- **Remote PSC Signaling:** In addition to receiving events from local trigger logic, the PSC FSM logic also receives and processes PSC signaling messages from the remote LER. Remote messages indicate the status of the transport path from the viewpoint of the far end LER. These messages may drive state changes on the local entity.
- **PSC Message Generator:** Based on the action output from the PSC control logic, this functional block formats the PSC protocol message and transmits it to the remote endpoint of the protected domain. This message may either be the same as the previously transmitted message or change when the PSC control has changed. The messages are transmitted as an initial burst followed by a regular interval.
- **Wait-to-Restore Timer:** The (configurable) WTR timer is used to delay reversion to a normal state when recovering from a failure condition on the working path in revertive mode. The PSC FSM logic starts/stops the WTR timer based on internal conditions/state. When the WTR expires, it generates an event to drive the local trigger logic.
- **Remote Event Expire Timer:** The (configurable) remote-event-expire timer is used to clear the remote event after the timer is expired because of remote inactivity or fault in the protected LSP. When the remote event clear timer expires, it generates a remote event clear notification to the PSC FSM logic.

Interoperability With Proprietary Lockout

An emulated protection (emulated automatic protection switching (APS)) switching ensures synchronization between peer entities. The emulated APS uses link down indication (LDI)message (proprietary) extensions when a lockout command is issued on the working or protected LSP. This lockout command is known as emLockout. A lockout is mutually exclusive between the working and protected LSP. In other words, when the working LSP is locked, the protected LSP cannot be locked (and vice versa).

The emLockout message is sent on the specified channel from the endpoint on the LSP where the lockout command (working/protected) is issued. Once the lockout is cleared locally, a Wait-To-Restore (WTR) timer (configurable) is started and the remote end notified. The local peer continues to remain in lockout until a clear is received from the remote peer and the WTR timer has expired and only then the LSP is considered to be no longer locked out. In certain deployments, you use a large WTR timer to emulate a non-revertive behavior. This causes the protected LSP to continue forwarding traffic even after the lockout has been removed from the working LSP.

The PSC protocol as specified in RFC-6378 is incompatible with the emulated APS implementation in certain conditions. For example, PSC implements a priority scheme whereby a lockout of protection (LoP) is at a higher priority than a forced switch (FS) issued on a working LSP. When an FS is issued and cleared, PSC states that the switching must revert to the working LSP immediately. However, the emulated APS implementation starts a WTR timer and switches after the timer has expired.

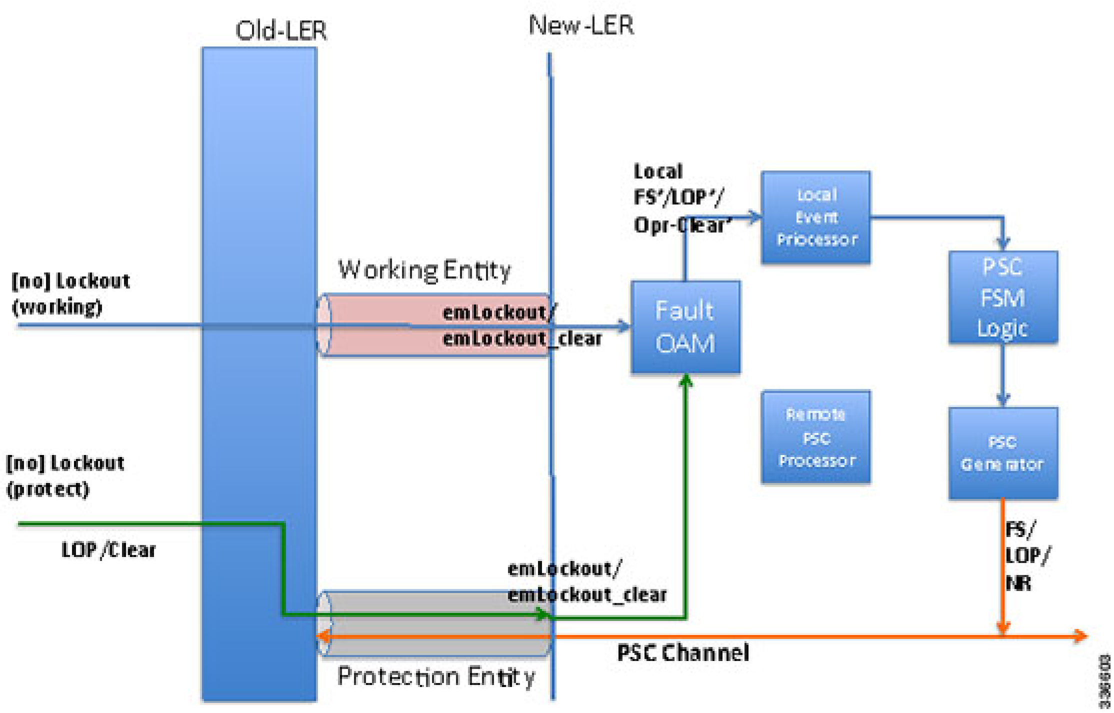
An endpoint implementing the newer PSC version may have to communicate with another endpoint implementing an older version. Because there is no mechanism to exchange the capabilities, the PSC implementation must interoperate with another peer endpoint implementing emulated APS. In this scenario, the new implementation sends both the LDI extension message (referred to as emLockout) as well as a PSC message when the lockout is issued.

Mapping and Priority of emlockout

There are two possible setups for interoperability:

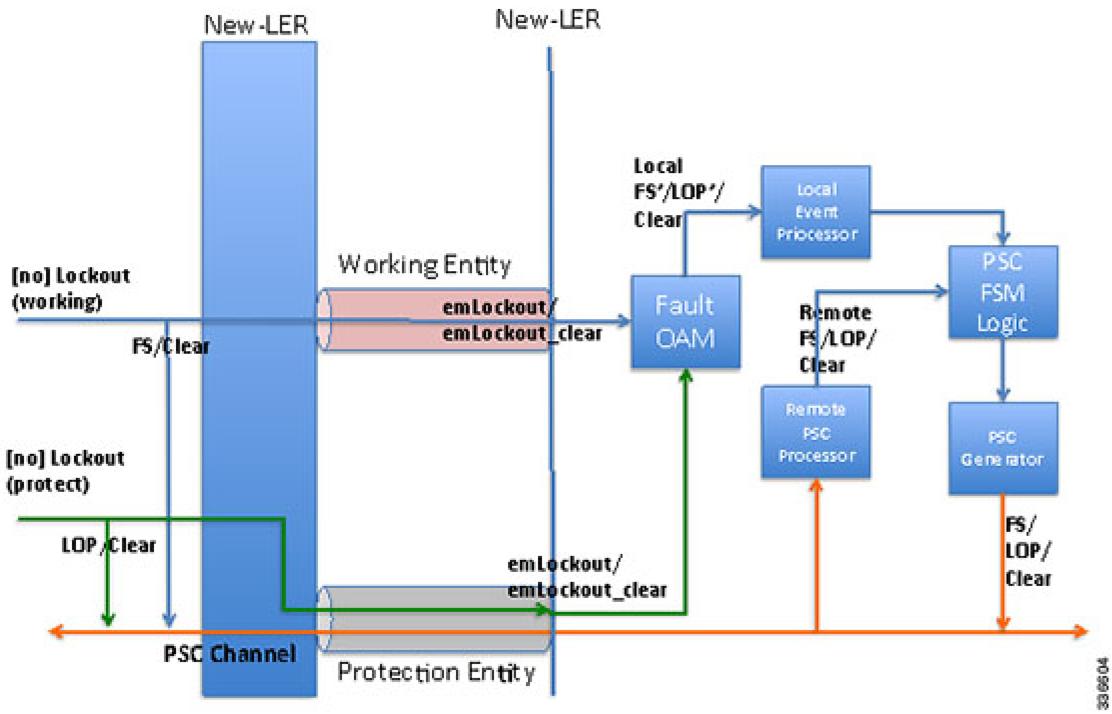
- New-old implementation.
- New-new implementation.

You can understand the mapping and priority when an emLockout is received and processed in the new-old implementation by referring to the following figure.



When the new label edge router (new-LER) receives an emLockout (or emLockout_clear) message, the new-LER maps the message into an internal local FS'/FSc' (local FS-prime/FS-prime-clear) or LoP'/LoPc' (local LoP-prime/Lop-prime-clear) event based on the channel on which it is received. This event is prioritized by the local event processor against any persistent local operator command. The highest priority event drives the PSC FSM logic and any associated path protection logic. A new internal state is defined for FS'/FSc' events. The PSC FSM logic transmits the corresponding PSC message. This message is dropped/ignored by the old-LER.

In the new-new LER implementation shown in the following figure, each endpoint generates two messages when a lockout command is given on a working or protected LSP.



When a lockout (working) command is issued, the new-LER implementation sends an emLockout command on the working LSP and PSC(FS) on the protected LSP. The remote peer receives two commands in either order. A priority scheme for local events is modified slightly beyond what is defined in order to drive the PSC FSM to a consistent state despite the order in which the two messages are received.

In the new implementation, it is possible to override the lockout of the working LSP with the lockout of the protected LSP according to the priority scheme. This is not allowed in the existing implementation. Consider the following steps between old (O) and new (N) node setup:

Time T1: Lockout (on the working LSP) is issued on O and N. Data is switched from the working to the protected LSP.

Time T2: Lockout (on the protected LSP) is issued on O and N. The command is rejected at O (existing behavior) and accepted at N (new behavior). Data in O->N continues on the protected LSP. Data in N->O switches to the working LSP.

You must issue a clear lockout (on the working LSP) and re-issue a lockout (on the protected LSP) on the old node to restore consistency.

WTR Synchronization

When a lockout on the working label switched path (LSP) is issued and subsequently cleared, a WTR timer (default: 10 sec, configurable) is started. When the timer expires, the data path is switched from protected to working LSP.

The PSC protocol indicates that the switch should happen immediately when a lockout (FS) is cleared.

When a new node is connected to the old node, for a period of time equal to the WTR timer value, the data path may be out-of-sync when a lockout is cleared on the working LSP. You should configure a low WTR value in order to minimize this condition.

Another issue is synchronization of the WTR value during stateful switchover (SSO). Currently, the WTR residual value is not checkpointed between the active and standby. As a result, after SSO, the new active restarts the WTR with the configured value if the protected LSP is active and the working LSP is up. As part of the PSC protocol implementation, the residual WTR is checkpointed on the standby. When the standby becomes active, the WTR is started with the residual value.

Priority of Inputs

The event priority scheme for locally generated events is as follows in high to low order:

Local Events:

1. Opr-Clear (Operator Clear)

2. LoP (Lockout of Protection)

3. LoP'/LoP'-Clear

4. FS (Forced Switch)

5. FS'/FS'-Clear

6. MS (Manual-Switch)

The emLockout received on the working LSP is mapped to the local-FS'. The emLockout received on the protected LSP is mapped to the local-LoP'. The emLockout-clear received is mapped to the corresponding clear events.

The priority definition for Signal Fail (SF), Signal Degrade (SD), Manual Switch (MS), WTR, Do Not Revert (DNR), and No Request (NR) remains unchanged.

PSC Syslogs

The following are the new syslogs that are introduced as part of the Linear Protection with PSC Support feature:

SYSLOG NAME	DESCRIPTION	RAW FORMAT
MPLS_TP_TUNNEL_PSC_PREEMPTION	Handle MPLS TP tunnel PSC event preemption syslog.	%MPLS-TP-5-PSCPREEMPTION: Tunnel-tp10, PSC Event: LOP:R preempted PSC Event: FS:L
MPLS_TP_TUNNEL_PSC_TYPE_MISMATCH	Handle MPLS TP tunnel type mismatch	%MPLS-PSC-5-TYPE-MISMATCH: Tunnel-tp10, type mismatch local-type: 1:1,

How to Configure MPLS Transport Profile

Configuring the MPLS Label Range

You must specify a static range of Multiprotocol Label Switching (MPLS) labels using the **mpls label range** command with the **static** keyword.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label range** *minimum-value maximum-value* **static** *minimum-static-value maximum-static-value*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls label range** *minimum-value maximum-value* **static** *minimum-static-value maximum-static-value* **Example:** `Device(config)# mpls label range 1001 1003 static 10000 25000`	Specifies a static range of MPLS labels.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring the Router ID and Global ID

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls tp**
4. **router-id** *node-id*
5. **global-id** *num*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls tp** **Example:** `Device(config)# mpls tp`	Enters MPLS-TP configuration mode, from which you can configure MPLS-TP parameters for the device.
Step 4	**router-id** *node-id* **Example:** `Device(config-mpls-tp)# router-id 10.10.10.10`	Specifies the default MPLS-TP router ID, which is used as the default source node ID for all MPLS-TP tunnels configured on the device.
Step 5	**global-id** *num* **Example:** `Device(config-mpls-tp)# global-id 1`	(Optional) Specifies the default global ID used for all endpoints and midpoints. • This command makes the router ID globally unique in a multiprovider tunnel. Otherwise, the router ID is only locally meaningful. • The global ID is an autonomous system number, which is a controlled number space by which providers can identify each other. • The router ID and global ID are also included in fault messages sent by devices from the tunnel midpoints to help isolate the location of faults.
Step 6	**end** **Example:** `Device(config-mpls-tp)# end`	Exits MPLS-TP configuration mode and returns to privileged EXEC mode.

Configuring Bidirectional Forwarding Detection Templates

The **bfd-template** command allows you to create a BFD template and enter BFD configuration mode. The template can be used to specify a set of BFD interval values. You invoke the template as part of the MPLS-TP tunnel. On platforms that support the BFD Hardware Offload feature and that can provide a 60-ms cutover for MPLS-TP tunnels, it is recommended to use the higher resolution timers in the BFD template.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **bfd-template single-hop** *template-name*
4. **interval** [**microseconds**] {**both** *time* | **min-tx** *time* **min-rx** *time*} [**multiplier** *multiplier-value*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bfd-template single-hop** *template-name* **Example:** `Device(config)# bfd-template single-hop mpls-bfd-1`	Creates a BFD template and enter BFD configuration mode.
Step 4	**interval** [**microseconds**] {**both** *time* | **min-tx** *time* **min-rx** *time*} [**multiplier** *multiplier-value*] **Example:** `Device(config-bfd)# interval min-tx 99 min-rx 99 multiplier 3`	Specifies a set of BFD interval values.
Step 5	**end** **Example:** `Device(config-bfd)# exit`	Exits BFD configuration mode and returns to privileged EXEC mode.

Configuring Pseudowire OAM Attributes

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-static-oam class** *class-name*
4. **timeout refresh send** *seconds*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-static-oam class** *class-name* **Example:** `Device(config)# pseudowire-static-oam class oam-class1`	Creates a pseudowire OAM class and enters pseudowire OAM class configuration mode.
Step 4	**timeout refresh send** *seconds* **Example:** `Device(config-st-pw-oam-class)# timeout refresh send 20`	Specifies the OAM timeout refresh interval.
Step 5	**exit** **Example:** `Device(config-st-pw-oam-class)# exit`	Exits pseudowire OAM configuration mode and returns to privileged EXEC mode.

Configuring the Pseudowire Class

When you create a pseudowire class, you specify the parameters of the pseudowire, such as the use of the control word, preferred path and OAM class template.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *class-name*
4. **encapsulation mpls**
5. **control-word**
6. **mpls label protocol** [ldp | none]
7. **preferred-path** {**interface tunnel** *tunnel-number* | **peer** {*ip-address* | *host-name*}} [**disable-fallback**]
8. **status protocol notification static** *class-name*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** *class-name* **Example:** `Device(config)# pseudowire-class mpls-tp-class1`	Creates a pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the encapsulation type.
Step 5	**control-word** **Example:** `Device(config-pw-class)# control-word`	Enables the use of the control word.
Step 6	**mpls label protocol** [ldp | none] **Example:** `Device(config-pw-class)#` **`protocol none`**	Specifies the type of protocol.
Step 7	**preferred-path** {**interface tunnel** *tunnel-number* | **peer** {*ip-address* | *host-name*}} [**disable-fallback**] **Example:** `Device(config-pw-class)# preferred-path interface tunnel-tp2`	Specifies the tunnel to use as the preferred path.
Step 8	**status protocol notification static** *class-name* **Example:** `Device(config-pw-class)# status protocol notification static oam-class1`	Specifies the OAM class to use.
Step 9	**end** **Example:** `Device(config-pw-class)# end`	Exits pseudowire class configuration mode and returns to privileged EXEC mode.

Configuring the Pseudowire

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface***interface-id*
4. **service instance** *number* **ethernet** [*name*]
5. **mpls label** *local-pseudowire-label remote-pseudowire-label*
6. **mpls control-word**
7. **backup delay** {*enable-delay-period* | **never**} {*disable-delay-period* | **never**}
8. **backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*] [**priority** *value*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface***interface-id* **Example:** `Router(config)# interface gigabitethernet 0/0/4`	Specifies the port on which to create the pseudowire and enters interface configuration mode. Valid interfaces are physical Ethernet ports.
Step 4	**service instance** *number* **ethernet** [*name*] **Example:** `Router(config-if)# service instance 2 ethernet`	Configure an EFP (service instance) and enter service instance configuration) mode. • *number*—Indicates EFP identifier. Valid values are from 1 to 400 • (Optional) **ethernet** *name*—Name of a previously configured EVC. You do not need to use an EVC name in a service instance. **Note** You can use service instance settings such as encapsulation, dot1q, and rewrite to configure tagging properties for a specific traffic flow within a given pseudowire session. For more information, see Ethernet Virtual Connections on the Cisco ASR 903 Router.

	Command or Action	Purpose
Step 5	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-if-xconn)# mpls label 1000 1001`	Configures the static pseudowire connection by defining local and remote circuit labels.
Step 6	**mpls control-word** **Example:** `Device(config-if-xconn)# no mpls control-word`	Specifies the control word.
Step 7	**backup delay** {*enable-delay-period* | **never**} {*disable-delay-period* | **never**} **Example:** `Device(config-if-xconn)# backup delay 0 never`	Specifies how long a backup pseudowire virtual circuit (VC) should wait before resuming operation after the primary pseudowire VC goes down.
Step 8	**backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*] [**priority** *value*] **Example:** `Device(config-if-xconn)# backup peer 10.0.0.2 50`	Specifies a redundant peer for a pseudowire virtual circuit (VC).
Step 9	**end** **Example:** `Device(config)# end`	Exits xconn interface connection mode and returns to privileged EXEC mode.

Configuring the MPLS-TP Tunnel

On the endpoint devices, create an MPLS TP tunnel and configure its parameters. See the interface tunnel-tp command for information on the parameters.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel-tp** *number*
4. **description** *tunnel-description*
5. **tp tunnel-name** *name*
6. **tp source** *node-id* [*global-id num*]
7. **tp destination** *node-id* [**tunnel-tp** *num*[**global-id** *num*]]
8. **bfd** *bfd-template*
9. **working-lsp**
10. **in-label** *num*
11. **out-label** *num* **out-link** *num*

12. **exit**
13. **protect-lsp**
14. **in-label** *num*
15. **out-label** *num* **out-link** *num*
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel-tp** *number* **Example:** `Device(config)# interface tunnel-tp 1`	Enters tunnel interface configuration mode. Tunnel numbers from 0 to 999 are supported.
Step 4	**description** *tunnel-description* **Example:** `Device(config-if)# description headend tunnel`	(Optional) Specifies a tunnel description.
Step 5	**tp tunnel-name** *name* **Example:** `Device(config-if)# tp tunnel-name tunnel 122`	Specifies the name of the MPLS-TP tunnel.
Step 6	**tp source** *node-id* [*global-id num*] **Example:** `Device(config-if)# tp source 10.11.11.11 global-id 10`	(Optional) Specifies the tunnel source and endpoint.
Step 7	**tp destination** *node-id* [**tunnel-tp** *num*[**global-id** *num*]] **Example:** `Device(config-if)# tp destination 10.10.10.10`	Specifies the destination node of the tunnel.
Step 8	**bfd** *bfd-template* **Example:**	Specifies the BFD template.

	Command or Action	Purpose
	`Device(config-if)# bfd mpls-bfd-1`	
Step 9	**working-lsp** **Example:** `Device(config-if)# working-lsp`	Specifies a working LSP, also known as the primary LSP.
Step 10	**in-label** *num* **Example:** `Device(config-if-working)# in-label 20000`	Specifies the in-label number.
Step 11	**out-label** *num* **out-link** *num* **Example:** `Device(config-if-working)# out-label 20000 out-link`	Specifies the out-label number and out-link.
Step 12	**exit** **Example:** `Device(config-if-working)# exit`	Exits working LSP interface configuration mode and returns to interface configuration mode.
Step 13	**protect-lsp** **Example:** `Device(config-if)# protect-lsp`	Specifies a backup for a working LSP.
Step 14	**in-label** *num* **Example:** `Device(config-if-protect)# in-label 20000`	Specifies the in label.
Step 15	**out-label** *num* **out-link** *num* **Example:** `Device(config-if-protect)# out-label 113 out-link`	Specifies the out label and out link.
Step 16	**end** **Example:** `Device(config-if-protect)# end`	Exits the interface configuration mode and returns to privileged EXEC mode.

Configuring MPLS-TP LSPs at Midpoints

Note When configuring LSPs at midpoint devices, ensure that the configuration does not deflect traffic back to the originating node.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls tp lsp source** *node-id* [**global-id** *num*] **tunnel-tp** *num* **lsp** {*lsp-num* | **protect** | **working**} **destination** *node-id* [**global-id** *num*] **tunnel-tp** *num*
4. **forward-lsp**
5. **in-label** *num* **out-label** *num* **out-link** *num*
6. **exit**
7. **reverse-lsp**
8. **in-label** *num* **out-label** *num* **out-link** *num*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls tp lsp source** *node-id* [**global-id** *num*] **tunnel-tp** *num* **lsp** {*lsp-num* | **protect** | **working**} **destination** *node-id* [**global-id** *num*] **tunnel-tp** *num* **Example:** `Device(config)# mpls tp lsp source 10.10.10.10 global-id 10 tunnel-tp 1 lsp protect destination 10.11.11.11 global-id 10 tunnel-tp 1`	Enables MPLS-TP midpoint connectivity and enters MPLS TP LSP configuration mode.
Step 4	**forward-lsp** **Example:** `Device(config-mpls-tp-lsp)# forward-lsp`	Enters MPLS-TP LSP forward LSP configuration mode.

	Command or Action	Purpose
Step 5	**in-label** *num* **out-label** *num* **out-link** *num* **Example:** `Device(config-mpls-tp-lsp-forw)# in-label 2000 out-label 2100 out-link 41`	Specifies the in label, out label, and out link numbers.
Step 6	**exit** **Example:** `Device(config-mpls-tp-lsp-forw)# exit`	Exits MPLS-TP LSP forward LSP configuration mode.
Step 7	**reverse-lsp** **Example:** `Device(config-mpls-tp-lsp)# reverse-lsp`	Enters MPLS-TP LSP reverse LSP configuration mode.
Step 8	**in-label** *num* **out-label** *num* **out-link** *num* **Example:** `Device(config-mpls-tp-lsp-rev)# in-label 22000 out-label 20000 out-link 44`	Specifies the in-label, out-label, and out-link numbers.
Step 9	**end** **Example:** `Device(config-mpls-tp-lsp-rev)# end`	Exits the MPLS TP LSP configuration mode and returns to privileged EXEC mode.

Configuring MPLS-TP Links and Physical Interfaces

MPLS-TP link numbers may be assigned to physical interfaces only. Bundled interfaces and virtual interfaces are not supported for MPLS-TP link numbers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **mpls tp link** *link-num*{**ipv4** *ip-address* **tx-mac** *mac-address*}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 1/0`	Specifies the interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.10.10.10 255.255.255.0`	Assigns an IP address to the interface.
Step 5	**mpls tp link** *link-num*{**ipv4** *ip-address* **tx-mac** *mac-address*} **Example:** `Device(config-if)# mpls tp link 1 ipv4 10.0.0.2`	Associates an MPLS-TP link number with a physical interface and next-hop node. On point-to-point interfaces or Ethernet interfaces designated as point-to-point using the **medium p2p**command, the next-hop can be implicit, so the **mpls tp link**command just associates a link number to the interface. Multiple tunnels and LSPs can refer to the MPLS-TP link to indicate they are traversing that interface. You can move the MPLS-TP link from one interface to another without reconfiguring all the MPLS-TP tunnels and LSPs that refer to the link. Link numbers must be unique on the device or node.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring MPLS-TP Linear Protection with PSC Support

The **psc** command allows you to configure MPLS-TP linear protection with PSC support. PSC is disabled by default. However, it can be enabled by issuing the **psc** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls tp**

4. **psc**
5. **psc fast refresh interval** *time-in-msec*
6. **psc slow refresh interval** *time-in-msec*
7. **psc remote refresh interval** *time-in-sec* **message-count** *num*
8. **exit**
9. **interface tunnel-tp** *number*
10. **psc**
11. **emulated-lockout**
12. **working-lsp**
13. **manual-switch**
14. **exit**
15. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls tp** **Example:** `Device(config)# mpls tp`	Enters Multiprotocol Label Switching (MPLS) Transport Profile (TP) global mode.
Step 4	**psc** **Example:** `Device(config-mpls-tp)# psc`	Enables the PSC Protocol.
Step 5	**psc fast refresh interval** *time-in-msec* **Example:** `Device(config-mpls-tp)# psc fast refresh interval 2000`	Configures the fast refresh interval for PSC messages. • The default is 1000 ms with a jitter of 50 percent. The range is from 1000 ms to 5000 sec.
Step 6	**psc slow refresh interval** *time-in-msec* **Example:** `Device(config-mpls-tp)# psc slow refresh interval 10`	Configures the slow refresh interval for PSC messages. • The default is 5 sec. The range is from 5 secs to 86400 secs (24 hours).

	Command or Action	Purpose
Step 7	**psc remote refresh interval** *time-in-sec* **message-count** *num* **Example:** `Device(config-mpls-tp)# psc remote refresh interval 20 message-count 15`	Configures the remote-event expiration timer. • By default, this timer is disabled. The remote refresh interval range is from 5 to 86400 sec (24 hours). The message count is from 5 to 1000. If you do not specify the message count value, it is set to 5, which is the default.
Step 8	**exit** **Example:** `Device(config-mpls-tp)# exit`	Exits MPLS TP global mode.
Step 9	**interface tunnel-tp** *number* **Example:** `Device(config)# interface tunnel-tp 1`	Creates an MPLS-TP tunnel called *number* and enters TP interface tunnel mode.
Step 10	**psc** **Example:** `Device(config-if)# psc`	Enables PSC. By default, PSC is disabled.
Step 11	**emulated-lockout** **Example:** `Device(config-if)# emulated-lockout`	Enables the sending of emLockout on working/protected transport entities if the **lockout** command is issued on each working/protected transport entity respectively. By default, the sending of emLockout is disabled.
Step 12	**working-lsp** **Example:** `Device(config-if)# working-lsp`	Enters working LSP mode on a TP tunnel interface.
Step 13	**manual-switch** **Example:** `Device(config-if-working)# manual-switch`	Issues a local manual switch condition on a working label switched path (LSP). This can be configured only in working LSP mode on a TP tunnel interface.
Step 14	**exit** **Example:** `Device(config-if-working)# exit`	Exits working LSP mode.
Step 15	**exit** **Example:** `Device(config-if)# exit`	Exits TP interface tunnel mode.

Configuring Static-to-Static Multisegment Pseudowires for MPLS-TP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *name* **point-to-point**
4. **bridge-domain***bridge-id*
5. **neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*}
6. **mpls label** *local-pseudowire-label remote-pseudowire-label*
7. **mpls control-word**
8. **neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*}
9. **mpls label** *local-pseudowire-label remote-pseudowire-label*
10. **mpls control-word**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2 vfi** *name* **point-to-point** **Example:** `Device(config)# l2 vfi atom point-to-point`	Creates a point-to-point Layer 2 virtual forwarding interface (VFI) and enters VFI configuration mode.
Step 4	**bridge-domain***bridge-id* **Example:** `Device)config)# bridge-domain 400`	Configures the bridge domain service instance. • *bridge-id*—Bridge domain identifier. The valid values are from 1 to 4000.
Step 5	**neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} **Example:** `Device(config-vfi)# neighbor 10.111.111.111 123 pw-class atom`	Sets up an emulated VC. Specify the IP address, the VC ID of the remote device, and the pseudowire class to use for the emulated VC. **Note** Only two **neighbor** commands are allowed for each Layer 2 VFI point-to-point command.

	Command or Action	Purpose
Step 6	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-vfi)# mpls label 10000 25000`	Configures the static pseudowire connection by defining local and remote circuit labels.
Step 7	**mpls control-word** **Example:** `Device(config-vfi)# mpls control-word`	Specifies the control word.
Step 8	**neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} **Example:** `Device(config-vfi)# neighbor 10.10.10.11 123 pw-class atom`	Sets up an emulated VC. Specify the IP address, the VC ID of the remote device, and the pseudowire class to use for the emulated VC.
Step 9	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-vfi)# mpls label 11000 11001`	Configures the static pseudowire connection by defining local and remote circuit labels.
Step 10	**mpls control-word** **Example:** **Example:** `Device(config-vfi)# mpls control-word`	Specifies the control word.
Step 11	**end** **Example:** `Device(config)# end`	Exits VFI configuration mode and returns to privileged EXEC mode.

Configuring Static-to-Dynamic Multisegment Pseudowires for MPLS-TP

When you configure static-to-dynamic pseudowires, you configure the static pseudowire class with the protocol none command, create a dynamic pseudowire class, and then invoke those pseudowire classes with the neighbor commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **pseudowire-class** *class-name*
4. **encapsulation mpls**
5. **control-word**
6. **mpls label protocol** [ldp | none]
7. **exit**
8. **pseudowire-class** *class-name*
9. **encapsulation mpls**
10. **exit**
11. **l2 vfi** *name* **point-to-point**
12. **neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*}
13. **neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*}
14. **mpls label** *local-pseudowire-label remote-pseudowire-label*
15. **mpls control-word**
16. **local interface** *pseudowire-type*
17. Do one of the following:
    - **tlv** [*type-name*] *type-value length* [**dec** | **hexstr** | **str**] *value*
    - **tlv template** *template-name*
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** *class-name* **Example:** `Device(config)# pseudowire-class mpls-tp-class1`	Creates a pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the encapsulation type.
Step 5	**control-word** **Example:** `Device(config-pw-class)# control-word`	Enables the use of the control word.

	Command or Action	Purpose
Step 6	**mpls label protocol** [ldp | none] **Example:** `Device(config-pw-class)# protocol none`	Specifies the type of protocol.
Step 7	**exit** **Example:** `Device(config-pw-class)# exit`	Exits pseudowire class configuration mode and returns to global configuration mode.
Step 8	**pseudowire-class** *class-name* **Example:** `Device(config)# pseudowire-class mpls-tp-class1`	Creates a pseudowire class and enters pseudowire class configuration mode.
Step 9	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the encapsulation type.
Step 10	**exit** **Example:** `Device(config-pw-class)# exit`	Exits pseudowire class configuration mode and returns to global configuration mode.
Step 11	**l2 vfi** *name* **point-to-point** **Example:** `Device(config)# l2 vfi atom point-to-point`	Creates a point-to-point Layer 2 virtual forwarding interface (VFI) and enters VFI configuration mode.
Step 12	**neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} **Example:** `Device(config-vfi)# neighbor 10.111.111.111 123 pw-class atom`	Sets up an emulated VC and enters VFI neighbor configuration mode. **Note** Note: Only two **neighbor** commands are allowed for each **l2 vfi point-to-point** command.
Step 13	**neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} **Example:** `Device(config-vfi-neighbor)# neighbor 10.111.111.111 123 pw-class atom`	Sets up an emulated VC. **Note** Only two **neighbor** commands are allowed for each **l2 vfi point-to-point** command.
Step 14	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:**	Configures the static pseudowire connection by defining local and remote circuit labels.

	Command or Action	Purpose
	`Device(config-vfi-neighbor)# mpls label 10000 25000`	
Step 15	**mpls control-word** **Example:** `Device(config-vfi-neighbor)# mpls control-word`	Specifies the control word.
Step 16	**local interface** *pseudowire-type* **Example:** `Device(config-vfi-neighbor)# local interface 4`	Specifies the pseudowire type.
Step 17	Do one of the following: • **tlv** [*type-name*] *type-value length* [**dec** | **hexstr** | **str**] *value* • **tlv template** *template-name* **Example:** `Device(config-vfi-neighbor)# tlv statictemp 2 4 hexstr 1`	Specifies the TLV parameters or invokes a previously configured TLV template.
Step 18	**end** **Example:** `Device(config-vfi-neighbor)# end`	Ends the session.

Configuring a Template with Pseudowire Type-Length-Value Parameters

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **tlv** [*type-name*] *type-value length* [**dec** | **hexstr** | **str**] *value*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**tlv** [*type-name*] *type-value length* [**dec** | **hexstr** | **str**] *value* **Example:** `Device(config-pw-tlv-template)# tlv statictemp 2 4  hexstr 1`	Specifies the TLV parameters.
Step 4	**end** **Example:** `Device(config-pw-tlv-template)# end`	Exits pseudowire TLV template configuration mode and returns to privileged EXEC mode.

Verifying the MPLS-TP Configuration

Use the following commands to verify and help troubleshoot your MPLS-TP configuration:

- **debug mpls tp**—Enables the logging of MPLS-TP error messages.
- **logging (MPLS-TP)**—Displays configuration or state change logging messages.
- **show bfd neighbors mpls-tp**—Displays the BFD state, which must be up in order for the endpoint LSPs to be up.
- **show mpls l2transport static-oam l2transport static-oam**—Displays MPLS-TP messages related to pseudowires.
- **show mpls tp tunnel-tp** *number* **detail**—Displays the number and details of the tunnels that are not functioning.
- **show mpls tp tunnel-tp lsps**—Displays the status of the LSPs, and helps you ensure that both LSPs are up and working from a tunnel endpoint.
- **traceroute mpls tp** and **ping mpls tp**—Helps you identify connectivity issues along the MPLS-TP tunnel path.

Configuration Examples for MPLS Transport Profile

Example: Configuring MPLS-TP Linear Protection with PSC Support

The following example enters MPLS TP global mode and enables the PSC Protocol.

```
Device> enable
Device# configure terminal
```

```
Device(config)# mpls tp
Device(config-mpls-tp)# psc
```

The following example configures the fast refresh interval for PSC messages. The interval value is 2000 seconds.

```
Device(config-mpls-tp)# psc fast refresh interval 2000
```

The following example configures the slow refresh interval for PSC messages. The interval value is 10 seconds.

```
Device(config-mpls-tp)# psc slow refresh interval 10
```

The following example configures the remote event expiration timer with a refresh interval value of 20 seconds with a message count of 15.

```
Device(config-mpls-tp)# psc remote refresh interval 20 message-count 15
```

The following example exits MPLS TP global mode, creates a TP interface tunnel, and enables PSC.

```
Device(config-mpls-tp)# exit
Decice(config) interface tunnel-tp 1
Device(config-if)# psc
```

The following example enables the sending of emLockout on working/protected transport entities, enters working LSP mode on a TP tunnel interface, and issues a local manual switch condition on a working LSP.

```
Device(config-if)# emulated-lockout
Device(config-if)# working-lsp
Device(config-if-working)# manual-switch
```

Example: Verifying MPLS-TP Linear Protection with PSC Support

The following example displays a summary of the MPLS-TP settings.

```
Device# show mpls tp summary
```

The following example provides information about the MPLS-TP link number database.

```
Device# show mpls tp link-numbers
```

Example: Troubleshooting MPLS-TP Linear Protection with PSC Support

The following example enables debugging for all PSC packets that are sent and received.

```
Device# debug mpls tp psc packet
```

The following example enables debugging for all kinds of PSC events.

```
Device# debug mpls tp psc event
```

The following example clears the counters for PSC signaling messages based on the tunnel number.

```
Device# clear mpls tp 1 psc counter
```

The following example clears the remote event for PSC based on the tunnel number.

```
Device# clear mpls tp tunnel-tp 1 psc remote-event
```

CHAPTER 3

MPLS Multilink PPP Support

> **Note** This chapter is not applicable on the ASR 900 RSP3 Module for the Cisco IOS XE Release 3.16.

The MPLS Multilink PPP Support feature ensures that MPLS Layer 3 Virtual Private Networks (VPNs) with quality of service (QoS) can be enabled for bundled links. This feature supports Multiprotocol Label Switching (MPLS) over Multilink PPP (MLP) links in the edge (provider edge [PE]-to-customer edge [CE]) or in the MPLS core (PE-to-PE and PE-to-provider [P] device).

Service providers that use relatively low-speed links can use MLP to spread traffic across them in their MPLS networks. Link fragmentation and interleaving (LFI) should be deployed in the CE-to-PE link for efficiency, where traffic uses a lower link bandwidth (less than 768 kbps). The MPLS Multilink PPP Support feature can reduce the number of Interior Gateway Protocol (IGP) adjacencies and facilitate load sharing of traffic.

Prerequisites for MPLS Multilink PPP Support

- Multiprotocol Label Switching (MPLS) must be enabled on provider edge (PE) and provider (P) devices

Restrictions for MPLS Multilink PPP Support

- Only 168 multilink bundles can be created per the OC-3 interface module on the router.
- The maximum number of members per multilink bundle is 16.
- Links in multilink bundles must be on the same interface module.
- On the 8 T1/E1, a maximum of 8 bundles can be supported.
- On the 16T1/E1, a maximum of 16 bundles can be supported.

- On the 32 T1/E1, a maximum of 32 bundles can be supported.

For information on how to configure, Protocol-Field-Compression (PFC) and Address-and-Control-Field-Compression (AFC), see *Configuring PPP and Multilink PPP on the Cisco ASR 903 Router*.

Information About MPLS Multilink PPP Support

MPLS Layer 3 Virtual Private Network Features Supported for Multilink PPP

The table below lists Multiprotocol Label Switching (MPLS) Layer 3 Virtual Private Network (VPN) features supported for Multilink PPP (MLP) and indicates if the feature is supported on customer edge-to-provider edge (CE-to-PE) links, PE-to-provider (P) links, and Carrier Supporting Carrier (CSC) CE-to-PE links.

Table 2: MPLS Layer 3 VPN Features Supported for MLP

MPLS L3 VPN Feature	CE-to-PE Links	PE-to-P Links	CSC CE-to-PE Links
Static routes	Supported	Not supported	Not supported
External Border Gateway Protocol (eBGP)	Supported	Not applicable to this configuration	Supported
Intermediate System-to-Intermediate System (IS-IS)	Not supported	Supported	Not supported
Open Shortest Path First (OSPF)	Supported	Supported	Not supported
Enhanced Interior Gateway Routing Protocol (EIGRP)	Supported	Supported	Not supported
Interprovider interautonomous (Inter-AS) VPNs (with Label Distribution Protocol [LDP])	Not applicable to this configuration	Supported (MLP between Autonomous System Boundary Routers [ASBRs])	Not applicable to this configuration
Inter-AS VPNs with IPv4 Label Distribution	Not applicable to this configuration	Supported (MLP between ASBRs)	Not applicable to this configuration
CSC VPNs (with LDP)	Not supported	Not applicable to this configuration	Supported
CSC VPNs with IPv4 label distribution	Supported	Not applicable to this configuration	Supported
External and internal BGP (eiBGP) Multipath	Not supported	Not supported	Not applicable to this configuration
Internal BGP (iBGP) Multipath	Not applicable to this configuration	Not supported	Not applicable to this configuration

MPLS L3 VPN Feature	CE-to-PE Links	PE-to-P Links	CSC CE-to-PE Links
eBGP Multipath	Not supported	Not supported	Not supported

MPLS Quality of Service Features Supported for Multilink PPP

The table below lists the Multiprotocol Label Switching (MPLS) quality of service (QoS) features supported for Multilink PPP (MLP) and indicates if the feature is supported on customer edge-to-provider edge (CE-to-PE) links, PE-to-provider (P) links, and Carrier Supporting Carrier (CSC) CE-to-PE links.

Table 3: MPLS QoS Features Supported for MLP

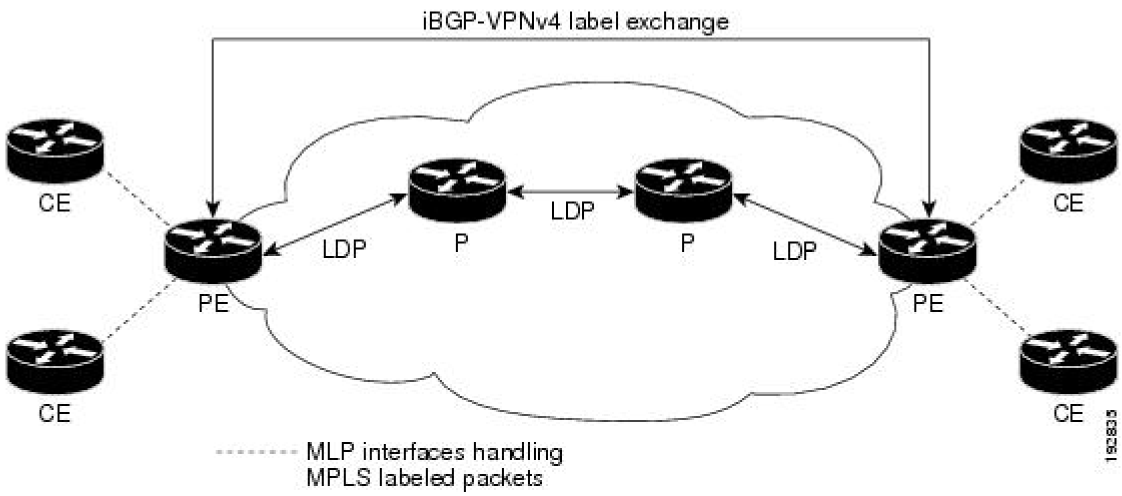
MPLS QoS Feature	CE-to-PE Links	PE-to-P Links	CSC CE-to-PE Links
Default copy of IP Precedence to EXP bits and the reverse	Supported	Not supported	Not supported
Set MPLS EXP bits using the modular QoS Command-Line Interface (MQC)	Supported	Supported	Supported
Matching on MPLS EXP using MQC	Supported	Supported	Supported
Low Latency Queueing (LLQ)/Class-Based Weighted Fair Queueing (CBWFQ) support	Supported	Supported	Supported
Weighted Random Early Detection (WRED) based on EXP bits using MQC	Supported	Supported	Supported
Policer with EXP bit-marking using MQC-3 action	Supported	Supported	Supported
Support for EXP bits in MPLS accounting	Supported	Supported	Supported

MPLS Multilink PPP Support and PE-to-CE Links

The figure below shows a typical Multiprotocol Label Switching (MPLS) network in which the provider edge (PE) device is responsible for label imposition (at ingress) and disposition (at egress) of the MPLS traffic.

In this topology, Multilink PPP (MLP) is deployed on the PE-to-customer edge (CE) links. The Virtual Private Network (VPN) routing and forwarding instance (VRF) interface is in a multilink bundle. There is no MPLS interaction with MLP; all packets coming into the MLP bundle are IP packets.

Figure 1: MLP and Traditional PE-to-CE Links



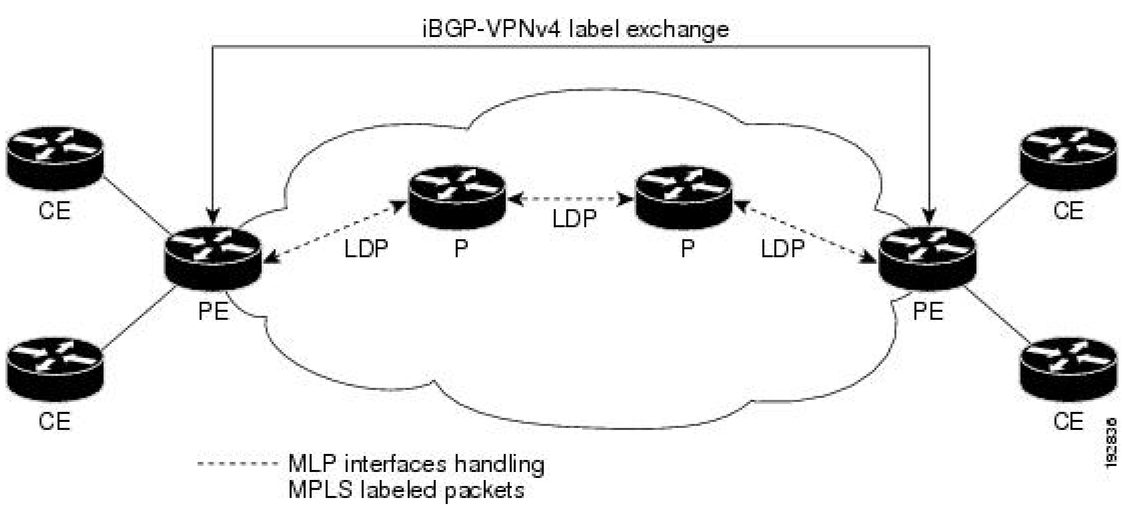
The PE-to-CE routing protocols that are supported for the MPLS Multilink PPP Support feature are external Border Gateway Protocol (eBGP), Open Shortest Path First (OSPF), and Enhanced Interior Gateway Routing Protocol (EIGRP). Static routes are also supported between the CE and PE devices.

Quality of service (QoS) features that are supported for the MPLS Multilink PPP Support feature on CE-to-PE links are link fragmentation and interleaving (LFI), compressed Real-Time Transport Protocol (cRTP), policing, marking, and classification.

MPLS Multilink PPP Support and Core Links

The figure below shows a sample topology in which Multiprotocol Label Switching (MPLS) is deployed over Multilink PPP (MLP) on provider edge-to-provider (PE-to-P) and P-to-P links. Enabling MPLS on MLP for PE-to-P links is similar to enabling MPLS on MLP for P-to-P links.

Figure 2: MLP on PE-to-P and P-to-P Links



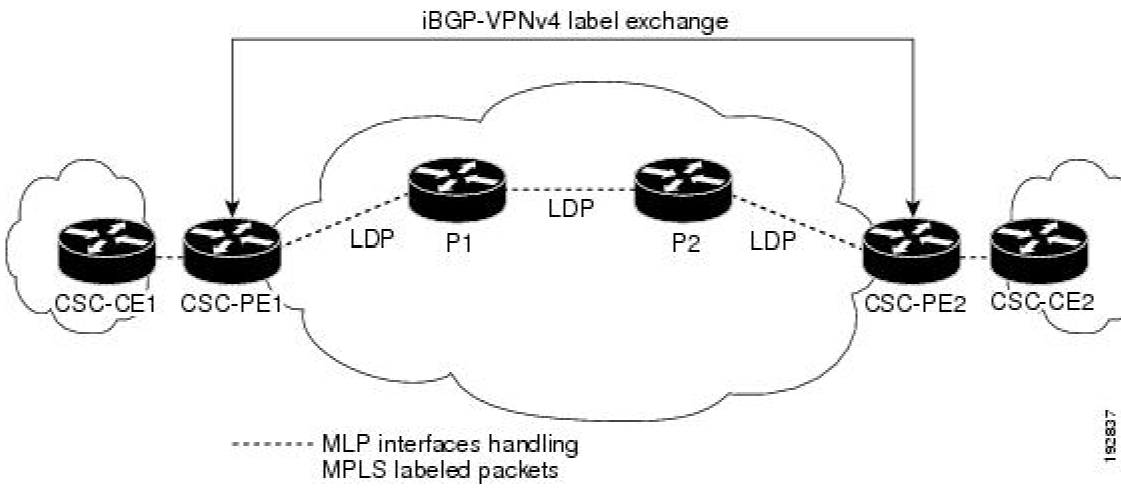
You employ MLP in the PE-to-P or P-to-P links primarily so that you can reduce the number of Interior Gateway Protocol (IGP) adjacencies and facilitate the load sharing of traffic.

In addition to requiring MLP on the PE-to-P links, the MPLS Multilink PPP Support feature requires the configuration of an IGP routing protocol and the Label Distribution Protocol (LDP).

MPLS Multilink PPP Support in a CSC Network

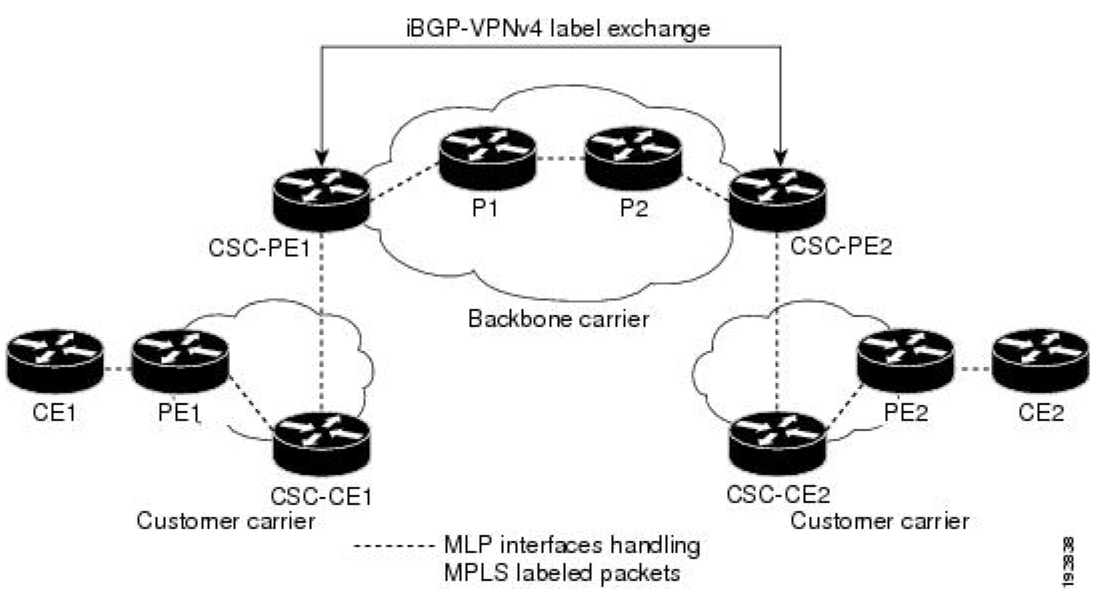
The figure below shows a typical Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) Carrier Supporting Carrier (CSC) network where Multilink PPP (MLP) is configured on the CSC customer edge (CE)-to-provider edge (PE) links.

Figure 3: MLP on CSC CE-to-PE Links with MPLS VPN Carrier Supporting Carrier



The MPLS Multilink PPP Support feature supports MLP between CSC-CE and CSC-PE links with the Label Distribution Protocol (LDP) or with external Border Gateway Protocol (eBGP) IPv4 label distribution. This feature also supports link fragmentation and interleaving (LFI) for an MPLS VPN CSC configuration. The figure below shows all MLP links that this feature supports for CSC configurations.

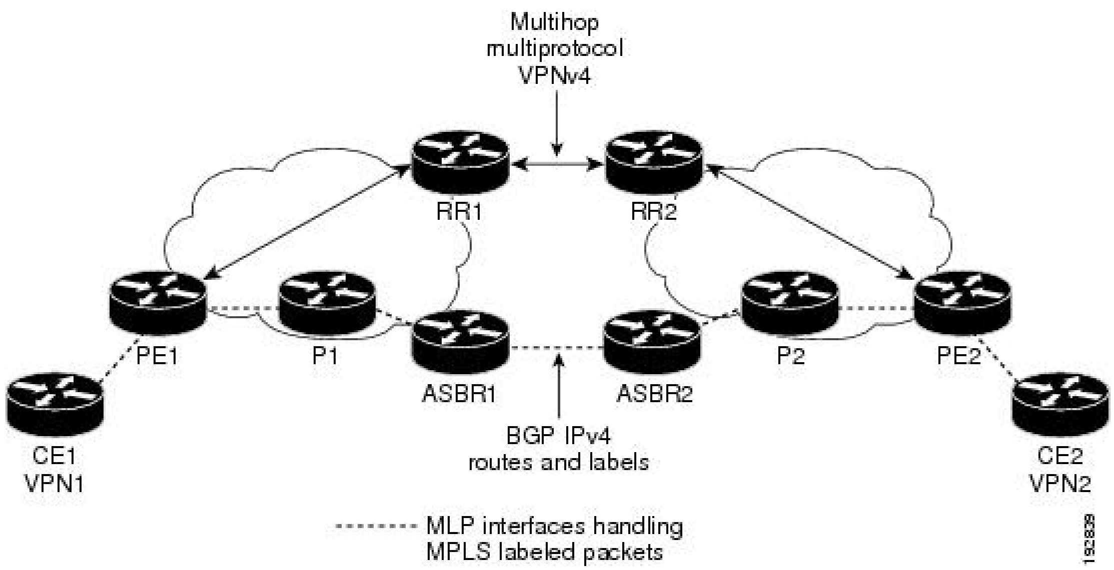
Figure 4: MLP Supported Links with MPLS VPN Carrier Supporting Carrier



MPLS Multilink PPP Support in an Interautonomous System

The figure below shows a typical Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) interautonomous system (Inter-AS) network where Multilink PPP (MLP) is configured on the provider edge-to-customer edge (PE-to-CE) links.

Figure 5: MLP on ASBR-to-PE Links in an MPLS VPN Inter-AS Network



The MPLS Multilink PPP Support feature supports MLP between Autonomous System Boundary Router (ASBR) links for Inter-AS VPNs with Label Distribution Protocol (LDP) and with external Border Gateway Protocol (eBGP) IPv4 label distribution.

How to Configure MPLS Multilink PPP Support

The tasks in this section can be performed on customer edge-to-provider edge (CE-to-PE) links, PE-to-provider (P) links, P-to-P links, and Carrier Supporting Carrier (CSC) CE-to-PE links.

Creating a Multilink Bundle

Perform this task to create a multilink bundle for the MPLS Multilink PPP Support feature. This multilink bundle can reduce the number of Interior Gateway Protocol (IGP) adjacencies and facilitate load sharing of traffic.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *group-number*
4. **ip address** *address mask* [**secondary**]
5. **encapsulation** *encapsulation-type*
6. **ppp multilink**

7. **mpls ip**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** `Device(config)# interface multilink 1`	Creates a multilink bundle and enters multilink interface configuration mode. • The *group-number* argument is the number of the multilink bundle (a nonzero number).
Step 4	**ip address** *address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.0 255.255.0.0`	Sets a primary or secondary IP address for an interface. • The *address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address. This command is used to assign an IP address to the multilink interface.
Step 5	**encapsulation** *encapsulation-type* **Example:** `Device(config-if)# encapsulation ppp`	Sets the encapsulation method as PPP to be used by the interface. • The *encapsulation-type* argument specifies the encapsulation type.
Step 6	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables MLP on an interface.
Step 7	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables label switching on the interface.

	Command or Action	Purpose
Step 8	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Assigning an Interface to a Multilink Bundle

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **controller** {**t1** \| **e1**} *slot/port*
4. **channel-group** *channel-number* **timeslots** *fulltimeslots*
5. **exit**
6. **interface serial** *slot/subslot* / *port* **:** *channel-group*
7. **ip route-cache** [**cef** | **distributed**]
8. **no ip address**
9. **keepalive** [*period* [*retries*]]
10. **encapsulation** *encapsulation-type*
11. **ppp multilink group** *group-number*
12. **ppp multilink**
13. **ppp authentication chap**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**controller** {**t1** | **e1**} *slot/port* **Example:** `Device# controller t1 0/0/1`	Configures a T1 or E1 controller and enters controller configuration mode. • The **t1** keyword indicates a T1 line card. • The **e1** keyword indicates an E1 line card. • The *slot/port* arguments are the backplane slot number and port number on the interface. Refer to your

	Command or Action	Purpose
		hardware installation manual for the specific slot numbers and port numbers.
Step 4	**channel-group** *channel-number* **timeslots** *fulltimeslots* **Example:** `Device(config-controller)# channel-group 1 timeslots 1-24`	Defines the time slots that belong to each T1 or E1 circuit. • The *channel-number* argument is the channel-group number. When a T1 data line is configured, channel-group numbers can be values from 1 to 24. When an E1 data line is configured, channel-group numbers can be values from 1 to 31. • The **timeslots** *fulltimeslots* keyword and argument specifies time slots. For a T1 controller, the time slot is 1-24. For an E1 controller the time slot is 1-31.
Step 5	**exit** **Example:** `Device(config-controller)# exit`	Returns to global configuration mode.
Step 6	**interface serial** *slot/subslot* / *port* **:** *channel-group* **Example:** `Device(config)# interface serial 0/0/1:1`	Configures a serial interface for a Cisco 7500 series router with channelized T1 or E1 and enters interface configuration mode. • The *slot* argument indicates the slot number. Refer to the appropriate hardware manual for slot and port information. • The */port* argument indicates the port number. Refer to the appropriate hardware manual for slot and port information. • The *:channel-group* argument indicates the channel group number. Cisco 7500 series routers specify the channel group number in the range of 0 to 4 defined with the **channel-group** controller configuration command.
Step 7	**ip route-cache** [**cef** | **distributed**] **Example:** `Device(config-if)# ip route-cache cef`	Controls the use of switching methods for forwarding IP packets. • The **cef** keyword enables Cisco Express Forwarding operation on an interface after Cisco Express Forwarding operation was disabled. • The **distributed** keyword enables distributed switching on the interface.
Step 8	**no ip address** **Example:**	Removes any specified IP address.

	Command or Action	Purpose
	`Device(config-if)# no ip address`	
Step 9	**keepalive** [*period* [*retries*]] **Example:** `Device(config-if)# keepalive`	Enables keepalive packets and specifies the number of times that the Cisco software tries to send keepalive packets without a response before bringing down the interface or before bringing the tunnel protocol down for a specific interface. • The *period* argument is an integer value, in seconds, greater than 0. The default is 10. • The *retries* argument specifies the number of times that the device continues to send keepalive packets without a response before bringing the interface down. Enter an integer value greater than 1 and less than 255. If you do not enter a value, the value that was previously set is used; if no value was specified previously, the default of 5 is used. If you are using this command with a tunnel interface, the command specifies the number of times that the device continues to send keepalive packets without a response before bringing the tunnel interface protocol down.
Step 10	**encapsulation** *encapsulation-type* **Example:** `Device(config-if)# encapsulation ppp`	Sets the encapsulation method used by the interface. • The *encapsulation-type* argument specifies the encapsulation type. The example specifies PPP encapsulation.
Step 11	**ppp multilink group** *group-number* **Example:** `Device(config-if)# ppp multilink group 1`	Restricts a physical link to join only one designated multilink group interface. • The *group-number* argument is the number of the multilink bundle (a nonzero number).
Step 12	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables MLP on the interface.
Step 13	**ppp authentication chap** **Example:** `Device(config-if)# ppp authentication chap`	(Optional) Enables Challenge Handshake Authentication Protocol (CHAP) authentication on the serial interface.
Step 14	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying the Multilink PPP Configuration

SUMMARY STEPS

1. **enable**
2. **show ip interface brief**
3. **show ppp multilink**
4. **show ppp multilink interface** *interface-bundle*
5. **show interface** *type number*
6. **show mpls forwarding-table**
7. **exit**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
Device#
```

Step 2 **show ip interface brief**

Verifies logical and physical Multilink PPP (MLP) interfaces.

Example:

```
Device# show ip interface brief

Locolrface                    IP-Address      OK? Method Status                Prot
GigabitEthernet1/0/0          10.3.62.106      YES NVRAM  up                    up
GigabitEthernet0/0/1          unassigned      YES NVRAM  administratively down down
GigabitEthernet0/0/0          unassigned      YES NVRAM  administratively down down
GigabitEthernet0/0/1          unassigned      YES NVRAM  administratively down down
GigabitEthernet0/0/2          unassigned      YES NVRAM  administratively down down
GigabitEthernet0/1/0          unassigned      YES NVRAM  administratively down down
GigabitEthernet0/1/1          unassigned      YES NVRAM  administratively down down
GigabitEthernet0/1/2          unassigned      YES NVRAM  administratively down down
Serial0/1/0:1                unassigned      YES NVRAM  administratively down down
Serial0/1/0:2                unassigned      YES NVRAM  administratively down down
Serial0/1/1:1                unassigned      YES NVRAM  up                    up
Serial0/1/1:2                unassigned      YES NVRAM  up                    down
Serial0/1/3:1                unassigned      YES NVRAM  up                    up
Serial0/1/3:2                unassigned      YES NVRAM  up                    up
Multilink6                   10.30.0.2       YES NVRAM  up                    up
Multilink8                   unassigned      YES NVRAM  administratively down down
Multilink10                  10.34.0.2       YES NVRAM  up                    up
Loopback0                    10.0.0.1        YES NVRAM  up                    up
```

Step 3 **show ppp multilink**

Verifies that you have created a multilink bundle.

Example:

```
Device# show ppp multilink

Multilink1, bundle name is group 1
 Bundle is Distributed
0 lost fragments, 0 reordered, 0 unassigned, sequence 0x0/0x0 rcvd/sent
0 discarded, 0 lost received, 1/255 load
Member links: 4 active, 0 inactive (max no set, min not set)
 Serial0/0/0/:1
 Serial0/0/0/:2
 Serial0/0/0/:3
 Serial0/0/0/:4
```

Step 4 **show ppp multilink interface** *interface-bundle*

Displays information about a specific MLP interface.

Example:

```
Device# show ppp multilink interface multilink6

Multilink6, bundle name is router
  Bundle up for 00:42:46, 1/255 load
  Receive buffer limit 24384 bytes, frag timeout 1524 ms
  Bundle is Distributed
    0/0 fragments/bytes in reassembly list
    1 lost fragments, 48 reordered
    0/0 discarded fragments/bytes, 0 lost received
    0x4D7 received sequence, 0x0 sent sequence
  Member links: 2 active, 0 inactive (max not set, min not set)
    Se0/1/3:1, since 00:42:46, 240 weight, 232 frag size
    Se0/1/3:2, since 00:42:46, 240 weight, 232 frag size
```

Step 5 **show interface** *type number*

Displays information about serial interfaces in your configuration.

Example:

```
Device# show interface serial 0/1/3:1

Serial0/1/3:1 is up, line protocol is up
  Hardware is Multichannel T1
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Open, multilink Open, crc 16, Data non-inverted
  Last input 00:00:01, output 00:00:01, output hang never
  Last clearing of "show interface" counters 00:47:13
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     722 packets input, 54323 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     697 packets output, 51888 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions no alarm present
  Timeslot(s) Used:1, subrate: 64Kb/s, transmit delay is 0 flags
  Transmit queue length 25
```

```
Device# show interface serial 0/1/3:2

Serial0/1/3:2 is up, line protocol is up
  Hardware is Multichannel T1
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Open, multilink Open, crc 16, Data non-inverted
  Last input 00:00:03, output 00:00:03, output hang never
  Last clearing of "show interface" counters 00:47:16
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     725 packets input, 54618 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     693 packets output, 53180 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions no alarm present
  Timeslot(s) Used:2, subrate: 64Kb/s, transmit delay is 0 flags
  Transmit queue length 26
```

You can also use the **show interface** command to display information about the multilink interface:

Example:

```
Device# show interface multilink6

Multilink6 is up, line protocol is up
  Hardware is multilink group interface
  Internet address is 10.30.0.2/8
  MTU 1500 bytes, BW 128 Kbit, DLY 100000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Open, multilink Open
  Open: CDPCP, IPCP, TAGCP, loopback not set
  DTR is pulsed for 2 seconds on reset
  Last input 00:00:00, output never, output hang never
  Last clearing of "show interface" counters 00:48:43
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     1340 packets input, 102245 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     1283 packets output, 101350 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Step 6 **show mpls forwarding-table**

Displays contents of the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB). Look for information on multilink interfaces associated with a point2point next hop.

Example:

```
Device# show mpls forwarding-table

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
```

```
tag    tag or VC   or Tunnel Id      switched   interface
16     Untagged    10.30.0.1/32      0          Mu6        point2point
17     Pop tag     10.0.0.3/32       0          Mu6        point2point
18     Untagged    10.0.0.9/32[V]    0          Mu10       point2point
19     Untagged    10.0.0.11/32[V]   6890       Mu10       point2point
20     Untagged    10.32.0.0/8[V]    530        Mu10       point2point
21     Aggregate   10.34.0.0/8[V]    0
22     Untagged    10.34.0.1/32[V]   0          Mu10       point2point
```

Use the **show ip bgp vpnv4** command to display VPN address information from the Border Gateway Protocol (BGP) table.

Example:

```
Device# show ip bgp vpnv4 all summary

BGP router identifier 10.0.0.1, local AS number 100
BGP table version is 21, main routing table version 21
10 network entries using 1210 bytes of memory
10 path entries using 640 bytes of memory
2 BGP path attribute entries using 120 bytes of memory
1 BGP extended community entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1994 total bytes of memory
BGP activity 10/0 prefixes, 10/0 paths, scan interval 5 secs
10.0.0.3  4 100 MsgRc52 MsgSe52  TblV21  0  0 00:46:35 State/P5xRcd
```

Step 7 **exit**

Returns to user EXEC mode.

Example:

```
Device# exit
Device>
```

Configuration Examples for MPLS Multilink PPP Support

Sample MPLS Multilink PPP Support Configurations

The following examples show sample configurations on a Carrier Supporting Carrier (CSC) network. The configuration of MLP on an interface is the same for provider edge-to-customer edge (PE-to-CE) links, PE-to-provider (P) links, and P-to-P links.

Example: Configuring Multilink PPP on an MPLS CSC PE Device

The following example shows how to configure for Multiprotocol Label Switching (MPLS) Carrier Supporting Carrier (CSC) provider edge (PE) device.

```
!
mpls label protocol ldp
ip cef
ip vrf vpn2
```

```
 rd 200:1
 route-target export 200:1
 route-target import 200:1
!
controller T1 0/0/1
framing esf
clock source internal
linecode b8zs
channel-group 1 timeslots 1-24
!
interface Serial0/0:1
 no ip address
 encapsulation ppp

 ppp multilink
 ppp multilink group 1

interface Multilink1
 ip vrf forwarding vpn2
 ip address 10.35.0.2 255.0.0.0
 no peer neighbor-route
 load-interval 30
 ppp multilink
 ppp multilink interleave
 ppp multilink group 1
mpls ip
mpls label protocol ldp
!
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Multilink1
 network 10.0.0.7 0.0.0.0 area 200
 network 10.31.0.0 0.255.255.255 area 200
!
!
 router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 10.0.0.11 remote-as 200
 neighbor 10.0.0.11 update-source Loopback0
 !
 address-family vpnv4
 neighbor 10.0.0.11 activate
 neighbor 10.0.0.11 send-community extended
 bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn2
 redistribute connected
 neighbor 10.35.0.1 remote-as 300
 neighbor 10.35.0.1 activate
 neighbor 10.35.0.1 as-override
 neighbor 10.35.0.1 advertisement-interval 5
 no auto-summary
 no synchronization
 exit-address-family
```

Example: Creating a Multilink Bundle

The following example shows how to create a multilink bundle for the MPLS Multilink PPP Support feature:

```
Device(config)# interface multilink 1
Device(config-if)# ip address 10.0.0.0 10.255.255.255
Device(config-if)# encapsulation ppp
Device(config-if)# ppp chap hostname group 1
Device(config-if)# ppp multilink
Device(config-if)# ppp multilink group 1
Device(config-if)# mpls ip
Device(config-if)# mpls label protocol ldp
```

Example: Assigning an Interface to a Multilink Bundle

The following example shows how to create four multilink interfaces with Cisco Express Forwarding switching and Multilink PPP (MLP) enabled. Each of the newly created interfaces is added to a multilink bundle.

```
interface multilink1
 ip address 10.0.0.0 10.255.255.255
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1
mpls ip
mpls label protocol ldp

interface serial 0/0/0/:1
 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp multilink
 ppp multilink group 1

 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1

 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1

 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1
```

CHAPTER 4

MPLS LSP Ping, Traceroute, and AToM VCCV

As Multiprotocol Label Switching (MPLS) deployments increase and the traffic types they carry increase, the ability of service providers to monitor label switched paths (LSPs) and quickly isolate MPLS forwarding problems is critical to their ability to offer services. The MPLS LSP Ping, Traceroute, and AToM VCCV feature helps them mitigate these challenges.

The MPLS LSP Ping, Traceroute, and AToM VCCV feature can detect when an LSP fails to deliver user traffic.

- You can use MPLS LSP Ping to test LSP connectivity for IPv4 Label Distribution Protocol (LDP) prefixes, traffic engineering (TE) Forwarding Equivalence Classes (FECs), and AToM FECs.
- You can use MPLS LSP Traceroute to trace the LSPs for IPv4 LDP prefixes and TE tunnel FECs.
- Any Transport over MPLS Virtual Circuit Connection Verification (AToM VCCV) allows you to use MPLS LSP Ping to test the pseudowire (PW) section of an AToM virtual circuit (VC).

Internet Control Message Protocol (ICMP) ping and trace are often used to help diagnose the root cause when a forwarding failure occurs. The MPLS LSP Ping, Traceroute, and AToM VCCV feature extends this diagnostic and troubleshooting ability to the MPLS network and aids in the identification of inconsistencies between the IP and MPLS forwarding tables, inconsistencies in the MPLS control and data plane, and problems with the reply path.

The MPLS LSP Ping, Traceroute, and AToM VCCV feature uses MPLS echo request and reply packets to test LSPs. The Cisco implementation of MPLS echo request and echo reply are based on the Internet Engineering Task Force (IETF) Internet-Draft *Detecting MPLS Data Plane Failures*.

Prerequisites for MPLS LSP Ping, Traceroute, and AToM VCCV

Before you use the MPLS LSP Ping, Traceroute, and AToM VCCV feature, you should:

- Determine the baseline behavior of your Multiprotocol Label Switching (MPLS) network. For example:
  - What is the expected MPLS experimental (EXP) treatment?
  - What is the expected maximum size packet or maximum transmission unit (MTU) of the label switched path?

    - What is the topology? What are the expected label switched paths? How many links in the label switching path (LSP)? Trace the paths of the label switched packets including the paths for load balancing.

- Understand how to use MPLS and MPLS applications, including traffic engineering, Any Transport over MPLS (AToM), and Label Distribution Protocol (LDP). You need to
    - Know how LDP is configured
    - Understand AToM concepts
    - Be able to troubleshoot a TE tunnel
- Understand label switching, forwarding, and load balancing.

Restrictions for MPLS LSP Ping, Traceroute, and AToM VCCV

- You cannot use MPLS LSP Traceroute to trace the path taken by Any Transport over Multiprotocol Label Switching (AToM) packets. MPLS LSP Traceroute is not supported for AToM. (MPLS LSP Ping is supported for AToM.) However, you can use MPLS LSP Traceroute to troubleshoot the Interior Gateway Protocol (IGP) LSP that is used by AToM.
- You cannot use MPLS LSP Ping or Traceroute to validate or trace MPLS Virtual Private Networks (VPNs).
- You cannot use MPLS LSP Traceroute to troubleshoot label switching paths (LSPs) that employ time-to-live (TTL) hiding.

Information About MPLS LSP Ping, Traceroute, and AToM VCCV

MPLS LSP Ping Operation

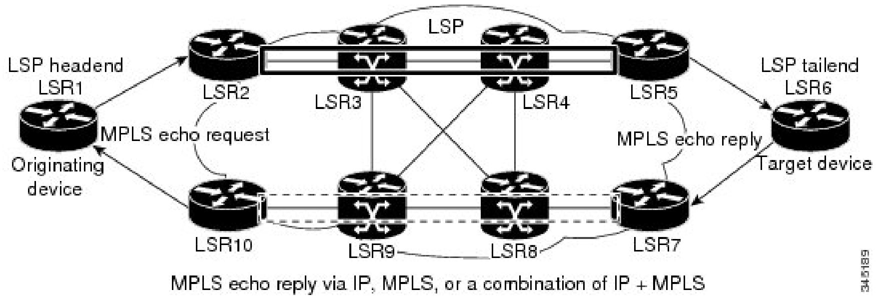
MPLS LSP Ping uses Multiprotocol Label Switching (MPLS) echo request and reply packets to validate a label switched path (LSP). Both an MPLS echo request and an MPLS echo reply are User Datagram Protocol (UDP) packets with source and destination ports set to 3503.

The MPLS echo request packet is sent to a target device 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 switched inband of the LSP (that is, 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 address of the UDP packet is defined as a 127.*x* .*y* .*z* /8 address. This 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. It is sent as an IP packet and forwarded using IP, MPLS, or a combination of both types of switching. The source address of the MPLS echo reply packet is an address from the device generating the echo reply. The destination address is the source address of the device in the MPLS echo request packet.

The figure below shows the echo request and echo reply paths for MPLS LSP Ping.

Figure 6: MPLS LSP Ping Echo Request and Echo Reply Paths



If you initiate an MPLS LSP Ping request at LSR1 to a Forwarding Equivalence Class (FEC), at LSR6, you get the results shown in the table below .

Table 4: MPLS LSP Ping Example

Step	Device	Action
1.	LSR1	Initiates an MPLS LSP Ping request for an FEC at the target device LSR6 and sends an MPLS echo request to LSR2.
1.	LSR2	Receives and forwards the MPLS echo request packet through transit devices LSR3 and LSR4 to the penultimate device LSR5.
1.	LSR5	Receives the MPLS echo request, pops the MPLS label, and forwards the packet to LSR6 as an IP packet.
1.	LSR6	Receives the IP packet, processes the MPLS echo request, and sends an MPLS echo reply to LSR1 through an alternate route.
1.	LSR7 to LSR10	Receive and forward the MPLS echo reply back toward LSR1, the originating device.
1.	LSR1	Receives the MPLS echo reply in response to the MPLS echo request.

You can use MPLS LSP Ping to validate IPv4 Label Distribution Protocol (LDP), Any Transport over MPLS (AToM), and IPv4 Resource Reservation Protocol (RSVP) FECs by using appropriate keywords and arguments with the command:

```
ping mpls
 {ipv4

destination-address destination-mask
 | pseudowire

ipv4-address
 vc-id

| traffic-eng

tunnel-interface tunnel-number
}
```

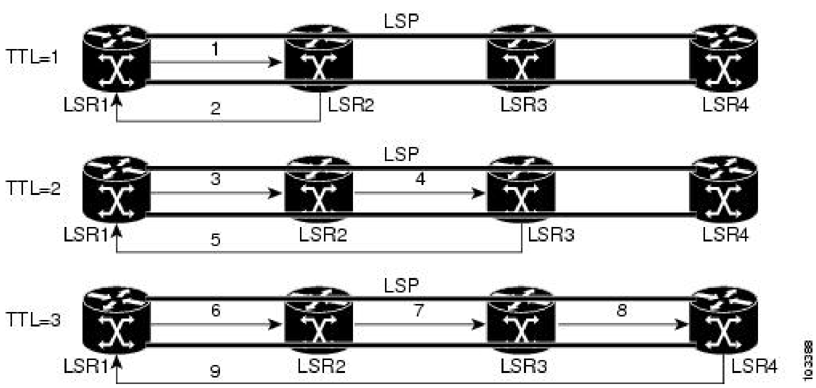
MPLS LSP Traceroute Operation

MPLS LSP Traceroute also uses Multiprotocol Label Switching (MPLS) echo request and reply packets to validate a label switched path (LSP). The echo request and echo reply are User Datagram Protocol (UDP) packets with source and destination ports set to 3503.

The MPLS LSP Traceroute feature uses time-to-live (TTL) settings to force expiration of the TTL along an LSP. MPLS LSP Traceroute incrementally increases the TTL value in its MPLS echo requests (TTL = 1, 2, 3, 4, ...) to discover the downstream mapping of each successive hop. The success of the LSP traceroute depends on the transit device processing the MPLS echo request when it receives a labeled packet with a TTL of 1. On Cisco devices, when the TTL expires, the packet is sent to the Route Processor (RP) for processing. The transit device returns an MPLS echo reply containing information about the transit hop in response to the TTL-expired MPLS packet.

The figure below shows an MPLS LSP Traceroute example with an LSP from LSR1 to LSR4.

Figure 7: MPLS LSP Traceroute Example



If you enter an LSP traceroute to a Forwarding Equivalence Class (FEC) at LSR4 from LSR1, you get the results shown in the table below.

Table 5: MPLS LSP Traceroute Example

Step	Device	MPLS Packet Type and Description	Device Action
1.	LSR1	MPLS echo request—With a target FEC pointing to LSR4 and to a downstream mapping.	• Sets the TTL of the label stack to 1. • Sends the request to LSR2.
1.	LSR2	MPLS echo reply.	Receives packet with TTL = 1. • Processes the UDP packet as an MPLS echo request. • Finds a downstream mapping, replies to LSR1 with its own downstream mapping based on the incoming label, and sends a reply.
1.	LSR1	MPLS echo request—With the same target FEC and the downstream mapping received in the echo reply from LSR2.	• Sets the TTL of the label stack to 2. • Sends the request to LSR2.

Step	Device	MPLS Packet Type and Description	Device Action
1.	LSR2	MPLS echo request.	Receives packet with TTL = 2. • Decrements the TTL. • Forwards the echo request to LSR3.
1.	LSR3	MPLS reply packet.	Receives packet with TTL = 1. • Processes the UDP packet as an MPLS echo request. • Finds a downstream mapping and replies to LSR1 with its own downstream mapping based on the incoming label.
1.	LSR1	MPLS echo request—With the same target FEC and the downstream mapping received in the echo reply from LSR3.	• Sets the TTL of the packet to 3. • Sends the request to LSR2.
1.	LSR2	MPLS echo request.	Receives packet with TTL = 3. • Decrements the TTL. • Forwards the echo request to LSR3.
1.	LSR3	MPLS echo request.	Receives packet with TTL = 2 • Decrements the TTL. • Forwards the echo request to LSR4.
1.	LSR4	MPLS echo reply.	Receives packet with TTL = 1. • Processes the UDP packet as an MPLS echo request. • Finds a downstream mapping and also finds that the device is the egress device for the target FEC. • Replies to LSR1.

You can use MPLS LSP Traceroute to validate IPv4 Label Distribution Protocol (LDP) and IPv4 RSVP FECs by using appropriate keywords and arguments with the **trace mpls** command:

```
trace mpls ipv4 {destination-address destination-mask | traffic-eng
 tunnel-interface tunnel-number}
```

By default, the TTL is set to 30. Therefore, the traceroute output always contains 30 lines, even if an LSP problem exists. This might mean duplicate entries in the output, should an LSP problem occur. The device address of the last point that the trace reaches is repeated until the output is 30 lines. You can ignore the duplicate entries. The following example shows that the trace encountered an LSP problem at the device that has an IP address of 10.6.1.6:

```
Device# traceroute mpls ipv4 10.6.7.4/32
Tracing MPLS Label Switched Path to 10.6.7.4/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
```

```
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
  0 10.6.1.14 MRU 4470 [Labels: 22 Exp: 0]
R 1 10.6.1.5 MRU 4470 [Labels: 21 Exp: 0] 2 ms
R 2 10.6.1.6 4 ms                  <------ Router address repeated for 2nd to 30th TTL.
R 3 10.6.1.6 1 ms
R 4 10.6.1.6 1 ms
R 5 10.6.1.6 3 ms
R 6 10.6.1.6 4 ms
R 7 10.6.1.6 1 ms
R 8 10.6.1.6 2 ms
R 9 10.6.1.6 3 ms
R 10 10.6.1.6 4 ms
R 11 10.6.1.6 1 ms
R 12 10.6.1.6 2 ms
R 13 10.6.1.6 4 ms
R 14 10.6.1.6 5 ms
R 15 10.6.1.6 2 ms
R 16 10.6.1.6 3 ms
R 17 10.6.1.6 4 ms
R 18 10.6.1.6 2 ms
R 19 10.6.1.6 3 ms
R 20 10.6.1.6 4 ms
R 21 10.6.1.6 1 ms
R 22 10.6.1.6 2 ms
R 23 10.6.1.6 3 ms
R 24 10.6.1.6 4 ms
R 25 10.6.1.6 1 ms
R 26 10.6.1.6 3 ms
R 27 10.6.1.6 4 ms
R 28 10.6.1.6 1 ms
R 29 10.6.1.6 2 ms
R 30 10.6.1.6 3 ms                  <------ TTL 30.
```

If you know the maximum number of hops in your network, you can set the TTL to a smaller value with the **trace mpls ttl** *maximum-time-to-live* command. The following example shows the same **traceroute** command as the previous example, except that this time the TTL is set to 5.

```
Device# traceroute mpls ipv4 10.6.7.4/32 ttl 5
Tracing MPLS Label Switched Path to 10.6.7.4/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
  0 10.6.1.14 MRU 4470 [Labels: 22 Exp: 0]
R 1 10.6.1.5 MRU 4474 [No Label] 3 ms
R 2 10.6.1.6 4 ms                  <------ Router address repeated for 2nd to 5th TTL.
R 3 10.6.1.6 1 ms
R 4 10.6.1.6 3 ms
R 5 10.6.1.6 4 ms
```

Any Transport over MPLS Virtual Circuit Connection Verification

AToM Virtual Circuit Connection Verification (AToM VCCV) allows the sending of control packets inband of an AToM pseudowire (PW) from the originating provider edge (PE) device. The transmission is intercepted at the destination PE device, instead of being forwarded to the customer edge (CE) device. This capability allows you to use MPLS LSP Ping to test the PW section of AToM virtual circuits (VCs).

AToM VCCV consists of the following:

- A signaled component in which the AToM VCCV capabilities are advertised during VC label signaling
- A switching component that causes the AToM VC payload to be treated as a control packet

AToM VCCV Signaling

One of the steps involved in Any Transport over Multiprotocol Label Switching (AToM) virtual circuit (VC) setup is the signaling of VC labels and AToM Virtual Circuit Connection Verification (VCCV) capabilities between AToM VC endpoints. The device uses an optional parameter, defined in the Internet Draft *draft-ieft-pwe3-vccv-01.txt,* to communicate the AToM VCCV disposition capabilities of each endpoint.

The AToM VCCV disposition capabilities are categorized as follows:

- Applications—MPLS LSP Ping and Internet Control Message Protocol (ICMP) Ping are applications that AToM VCCV supports to send packets inband of an AToM PW for control purposes.
- Switching modes—Type 1 and Type 2 are switching modes that AToM VCCV uses for differentiating between control and data traffic.

The table below describes AToM VCCV Type 1 and Type 2 switching modes.

Table 6: Type 1 and Type 2 AToM VCCV Switching Modes

Switching Mode	Description
Type 1	Uses a Protocol ID (PID) field in the AToM control word to identify an AToM VCCV packet.
Type 2	Uses an MPLS Router Alert Label above the VC label to identify an AToM VCCV packet.

Selection of AToM VCCV Switching Types

Cisco devices always use Type 1 switching, if available, when they send MPLS LSP Ping packets over an Any Transport over Multiprotocol Label Switching (AToM) virtual circuit (VC) control channel. Type 2 switching accommodates those VC types and implementations that do not support or interpret the AToM control word.

The table below shows the AToM Virtual Circuit Connection Verification (VCCV) switching mode advertised and the switching mode selected by the AToM VC.

Table 7: AToM VCCV Switching Mode Advertised and Selected by AToM Virtual Circuit

Type Advertised	Type Selected
AToM VCCV not supported	–
Type 1 AToM VCCV switching	Type 1 AToM VCCV switching
Type 2 AToM VCCV switching	Type 2 AToM VCCV switching
Type 1 and Type 2 AToM VCCV switching	Type 1 AToM VCCV switching

An AToM VC advertises its AToM VCCV disposition capabilities in both directions: that is, from the originating device (PE1) to the destination device (PE2), and from PE2 to PE1.

In some instances, AToM VCs might use different switching types if the two endpoints have different AToM VCCV capabilities. If PE1 supports Type 1 and Type 2 AToM VCCV switching and PE2 supports only Type 2 AToM VCCV switching, there are two consequences:

- LSP ping packets sent from PE1 to PE2 are encapsulated with Type 2 switching.
- LSP ping packets sent from PE2 to PE1 use Type 1 switching.

You can determine the AToM VCCV capabilities advertised to and received from the peer by entering the **show mpls l2transport binding** command at the PE device. For example:

```
Device# show mpls l2transport binding

  Destination Address: 10.131.191.252,  VC ID: 333
    Local Label:  16
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV Capabilities: Type 1, Type 2
    Remote Label: 19
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV Capabilities: Type 1
```

Command Options for ping mpls and trace mpls

MPLS LSP Ping and Traceroute command options are specified as keywords and arguments on the **ping mpls** and **trace mpls** commands.

The **ping mpls** command provides the options displayed in the command syntax below:

```
ping mpls ipv4{destination-address/destination-mask [destination address-start
address-end increment] [ttl time-to-live] | pseudowire ipv4-address
vc-id vc-id [destination address-start address-end increment] |
traffic-eng tunnel-interface tunnel-number [ttl time-to-live]} [source
source-address] [repeat count] [timeout seconds][{size
 packet-size} | {sweep minimum maximum size-Increment}] [pad pattern]
[reply mode {ipv4|router-alert}] [interval msec]
[exp exp-bits] [verbose]
```

The **trace mpls** command provides the options displayed in the command syntax below:

```
trace mpls {ipv4 destination-address/destination-mask [destination
address-start [address-end [address-increment]]] | traffic-eng tunnel tunnel-interface-number}
[source source-address] [timeout seconds] [reply mode reply-mode]
[ttl maximum-time-to-live] [exp exp-bits]
```

Selection of FECs for Validation

A label switched path (LSP) is formed by labels. Devices learn labels through the Label Distribution Protocol (LDP), traffic engineering (TE), Any Transport over Multiprotocol Label Switching (AToM), or other MPLS applications. You can use MPLS LSP Ping and Traceroute to validate an LSP used for forwarding traffic for a given Forwarding Equivalence Class (FEC). The table below lists the keywords and arguments for the **ping mpls** and **traceroute mpls** commands that allow the selection of an LSP for validation.

Table 8: Selection of LSPs for Validation

FEC Type	ping mpls Keyword and Argument	traceroute mpls Keyword and Argument
LDP IPv4 prefix	**ipv4** *destination-address destination-mask*	**ipv4** *destination-address destination-mask*
MPLS TE tunnel	**traffic-eng** *tunnel-interface tunnel-number* **Note** MPLS TE Tunnel is not applicable on the ASR 900 RSP3 Module for the Cisco IOS XE Release 3.16.	**traffic-eng** *tunnel-interface tunnel-number* **Note** MPLS TE Tunnel is not applicable on the ASR 900 RSP3 Module for the Cisco IOS XE Release 3.16.
AToM VC	**pseudowire** *ipv4-address* **vc-id** *vc-id*	MPLS LSP Traceroute does not support the AToM tunnel LSP type for this release.

Reply Mode Options for MPLS LSP Ping and Traceroute

The reply mode is used to control how the responding device replies to a Multiprotocol Label Switching (MPLS) echo request sent by an MPLS LSP Ping or MPLS LSP Traceroute command. The table below describes the reply mode options.

Table 9: Reply Mode Options for a Responding Device

Option	Description
ipv4	Reply with an IPv4 User Datagram Protocol (UDP) packet (default). This is the most common reply mode selected for use with an MPLS LSP Ping and Traceroute command when you want to periodically poll the integrity of a label switched path (LSP). With this option, you do not have explicit control over whether the packet traverses IP or MPLS hops to reach the originator of the MPLS echo request. If the headend device fails to receive a reply, select the router-alert option, "Reply with an IPv4 UDP packet with a router alert." The responding device sets the IP precedence of the reply packet to 6. You implement this option using the **reply mode ipv4** keywords.
router-alert	Reply with an IPv4 UDP packet with a device alert. This reply mode adds the router alert option to the IP header. This forces the packet to be special handled by the Cisco device at each intermediate hop as it moves back to the destination. This reply mode is more expensive, so use the router-alert option only if you are unable to get a reply with the ipv4 option, "Reply with an IPv4 UDP packet." You implement this option using the **reply mode router-alert** keywords

The reply with an IPv4 UDP packet implies that the device should send an IPv4 UDP packet in reply to an MPLS echo request. If you select the ipv4 reply mode, you do not have explicit control over whether the packet uses IP or MPLS hops to reach the originator of the MPLS echo request. This is the mode that you would normally use to test and verify LSPs.

The reply with an IPv4 UDP packet that contains a device alert forces the packet to go back to the destination and be processed by the Route Processor (RP) process switching at each intermediate hop. This bypasses hardware/line card forwarding table inconsistencies. You should select this option when the originating (headend) devices fail to receive a reply to the MPLS echo request.

You can instruct the replying device to send an echo reply with the IP router alert option by using one of the following commands:

```
ping mpls
 {ipv4 destination-address/destination-mask | pseudowire ipv4-address
 vc-idvc-id | traffic-engtunnel-interface tunnel-number}
reply mode  router-alert
```

or

```
trace mpls
 {ipv4destination-address/destination-mask
 | traffic-eng tunnel-interface tunnel-number
} reply mode router-alert
```

However, the reply with a router alert adds overhead to the process of getting a reply back to the originating device. This method is more expensive to process than a reply without a router alert and should be used only if there are reply failures. That is, the reply with a router alert label should only be used for MPLS LSP Ping or MPLS LSP Traceroute when the originating (headend) device fails to receive a reply to an MPLS echo request.

Reply Mode Options for MPLS LSP Ping and Traceroute

The reply mode is used to control how the responding device replies to a Multiprotocol Label Switching (MPLS) echo request sent by an MPLS LSP Ping or MPLS LSP Traceroute command. The table below describes the reply mode options.

Table 10: Reply Mode Options for a Responding Device

Option	Description
ipv4	Reply with an IPv4 User Datagram Protocol (UDP) packet (default). This is the most common reply mode selected for use with an MPLS LSP Ping and Traceroute command when you want to periodically poll the integrity of a label switched path (LSP). With this option, you do not have explicit control over whether the packet traverses IP or MPLS hops to reach the originator of the MPLS echo request. If the headend device fails to receive a reply, select the router-alert option, “Reply with an IPv4 UDP packet with a router alert.” The responding device sets the IP precedence of the reply packet to 6. You implement this option using the **reply mode ipv4** keywords.
router-alert	Reply with an IPv4 UDP packet with a device alert. This reply mode adds the router alert option to the IP header. This forces the packet to be special handled by the Cisco device at each intermediate hop as it moves back to the destination. This reply mode is more expensive, so use the router-alert option only if you are unable to get a reply with the ipv4 option, “Reply with an IPv4 UDP packet.” You implement this option using the **reply mode router-alert** keywords

The reply with an IPv4 UDP packet implies that the device should send an IPv4 UDP packet in reply to an MPLS echo request. If you select the ipv4 reply mode, you do not have explicit control over whether the packet uses IP or MPLS hops to reach the originator of the MPLS echo request. This is the mode that you would normally use to test and verify LSPs.

The reply with an IPv4 UDP packet that contains a device alert forces the packet to go back to the destination and be processed by the Route Processor (RP) process switching at each intermediate hop. This bypasses hardware/line card forwarding table inconsistencies. You should select this option when the originating (headend) devices fail to receive a reply to the MPLS echo request.

You can instruct the replying device to send an echo reply with the IP router alert option by using one of the following commands:

```
ping mpls
 {ipv4 destination-address/destination-mask | pseudowire ipv4-address
 vc-idvc-id | traffic-engtunnel-interface tunnel-number}
reply mode  router-alert
```

or

```
trace mpls
 {ipv4destination-address/destination-mask
 | traffic-eng tunnel-interface tunnel-number
} reply mode router-alert
```

However, the reply with a router alert adds overhead to the process of getting a reply back to the originating device. This method is more expensive to process than a reply without a router alert and should be used only if there are reply failures. That is, the reply with a router alert label should only be used for MPLS LSP Ping or MPLS LSP Traceroute when the originating (headend) device fails to receive a reply to an MPLS echo request.

Packet Handling Along Return Path with an IP MPLS Router Alert

When an IP packet that contains an IP router alert option in its IP header or a Multiprotocol Label Switching (MPLS) packet with a router alert label as its outermost label arrives at a device, the device punts (redirects) the packet to the Route Processor (RP) process level for handling. This allows these packets to bypass the forwarding failures in hardware routing tables. The table below describes how IP and MPLS packets with an IP router alert option are handled by the device switching path processes.

Table 11: Switching Path Process Handling of IP and MPLS Router Alert Packets

Incoming Packet	Normal Switching Action	Process Switching Action	Outgoing Packet
IP packet—Router alert option in IP header	A rRouter alert option in the IP header causes the packet to be punted to the process switching path.	Forwards the packet as is.	IP packet—Router alert option in IP header.
	A router alert option in theIP header causes the packet to be punted to the process switching path.	Adds a router alert as the outermost label and forwards as an MPLS packet.	MPLS packet— Outermost label contains a router alert.

Incoming Packet	Normal Switching Action	Process Switching Action	Outgoing Packet
MPLS packet—Outermost label contains a router alert	If the router alert label is the outermost label, the packet is punted to the process switching path.	Removes the outermost router alert label, adds an IP router alert option to the IP header, and forwards as an IP packet.	IP packet—Router alert option in IP header.
	If the router alert label is the outermost label, the packet is punted to the process switching path.	Preserves the outermost router alert label and forwards the MPLS packet.	MPLS packet— Outermost label contains a router alert.

Other MPLS LSP Ping and Traceroute Command Options

The table below describes other MPLS LSP Ping and Traceroute command options that can be specified as keywords or arguments with the **ping mpls** command, or with both the **ping mpls** and **trace mpls** commands. Options available to use only on the **ping mpls** command are indicated as such.

Table 12: Other MPLS LSP Ping and Traceroute and AToM VCCV Options

Option	Description
Datagram size	Size of the packet with the label stack imposed. Specified with the **size** *packet-size* keyword and argument. The default size is 100. For use with the MPLS LSP Ping feature only.
Padding	Padding (the pad time-length-value [TLV]) is used as required to fill the datagram so that the MPLS echo request (User Datagram Protocol [UDP] packet with a label stack) is the size specified. Specify with the **pad** *pattern* keyword and argument. For use with the MPLS LSP Ping feature only.
Sweep size range	Parameter that enables you to send a number of packets of different sizes, ranging from a start size to an end size. This parameter is similar to the Internet Control Message Protocol (ICMP) ping sweep parameter. The lower boundary on the sweep range varies depending on the label switched path (LSP) type. You can specify a sweep size range when you use the **ping mpls** command. Use the **sweep** *minimum maximum size-increment* keyword and arguments. For use with the MPLS LSP Ping feature only.
Repeat count	Number of times to resend the same packet. The default is 5 times. You can specify a repeat count when you use the **ping mpls** command. Use the **repeat** *count* keyword and argument. For use with the MPLS LSP Ping feature only.
MPLS echo request source address	Routable address of the sender. The default address is loopback0. This address is used as the destination address in the Multiprotocol Label Switching (MPLS) echo response. Use the **source** *source-address* keyword and argument. For use with the MPLS LSP Ping and Traceroute features.

Option	Description
UDP destination address	A valid 127/8 address. You have the option to specify a single *x.y.z* or a range of numbers between 0.0.0 and *x.y.z* , where *x.y.z* are numbers between 0 and 255 and correspond to 127.*x.y.z*. Use the **destination** {*address* | *address-start address-end increment*} keyword and arguments. The MPLS echo request destination address in the UDP packet is not used to forward the MPLS packet to the destination device. The label stack that is used to forward the echo request routes the MPLS packet to the destination device. The 127/8 address guarantees that the packets are routed to the localhost (the default loopback address of the device processing the address) if the UDP packet destination address is used for forwarding. In addition, the destination address is used to affect load balancing when the destination address of the IP payload is used for load balancing. For use with IPv4 and Any Transport over MPLS (AToM) Forwarding Equivalence Classes (FECs) with the MPLS LSP Ping feature and with IPv4 FECs with the MPLS LSP Traceroute feature.
Time-to-live (TTL)	A parameter you can set that indicates the maximum number of hops a packet should take to reach its destination. The time-to-live (TTL) field in a packet is decremented by 1 each time it travels through a device. For MPLS LSP Ping, the TTL is a value after which the packet is discarded and an MPLS echo reply is sent back to the originating device. Use the **ttl** *time-to-live* keyword and argument. For MPLS LSP Traceroute, the TTL is a maximum time to live and is used to discover the number of downstream hops to the destination device. MPLS LSP Traceroute incrementally increases the TTL value in its MPLS echo requests (TTL = 1, 2, 3, 4, ...) to accomplish this. Use the **ttl** *time-to-live* keyword and argument.
Timeouts	A parameter you can specify to control the timeout in seconds for an MPLS request packet. The range is from 0 to 3600 seconds. The default is 2. Set with the **timeout** *seconds* keyword and argument. For use with the MPLS LSP Ping and Traceroute features.
Intervals	A parameter you can specify to set the time in milliseconds between successive MPLS echo requests. The default is 0. Set with the **interval** *msec* keyword and argument.
Experimental bits	Three experimental bits in an MPLS header used to specify precedence for the MPLS echo reply. (The bits are commonly called EXP bits.) The range is from 0 to 7, and the default is 0. Specify with the **exp** *exp-bits* keyword and argument. For use with the MPLS LSP Ping and Traceroute features.

Option	Description
Verbose	Option that provides additional information for the MPLS echo reply--source address and return codes. For the MPLS LSP Ping feature, this option is implemented with the **verbose** keyword. For use with the MPLS LSP Ping feature only.

MPLS LSP Ping options described in the table above can be implemented by using the following syntax:

```
ping mpls
{ipv4 destination-address destination-mask [destination address-start address-end increment]

 [ttl time-to-live] | pseudowire ipv4-address
 vc-id vc-id
[destination address-start address-end increment] | traffic-eng tunnel-interface
tunnel-number
[ttl time-to-live]}
[source source-address] [repeat count]
[{size packet-size} | {sweep minimum maximum size-Increment}]
[pad pattern]
[timeout seconds] [intervalmsec]
[exp exp-bits] [verbose
```

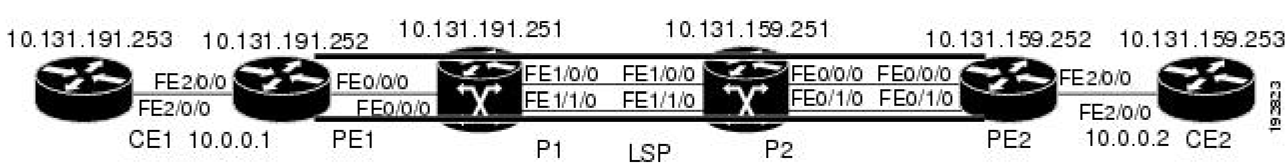
MPLS LSP Traceroute options described in the table below can be implemented by the use of the following syntax:

```
trace mpls
 {ipv4 destination-address destination-mask
 [destination address-start address-end address-increment] | traffic-eng tunnel-interface
tunnel-number}
[source source-address] [timeout seconds]
[ttl maximum-time-to-live]
[exp exp-bits]
```

Option Interactions and Loops

Usage examples for the MPLS LSP Ping and Traceroute and AToM VCCV feature in this and subsequent sections are based on the sample topology shown in the figure below.

Figure 8: Sample Topology for Configuration Examples



The interaction of some MPLS LSP Ping and Traceroute and AToM VCCV options can cause loops. See the following topic for a description of the loops you might encounter with the **ping mpls** and **trace mpls** commands:

Possible Loops with MPLS LSP Ping

With the MPLS LSP Ping feature, loops can occur if you use the repeat count option, the sweep size range option, or the User Datagram Protocol (UDP) destination address range option.

```
ping mpls
 {ipv4 destination-address/destination-mask
[destination address-start address-end increment] | pseudowire ipv4-address
 vc-id vc-id
 [destination address-start address-end increment] |
traffic-eng tunnel-interface tunnel-number}
[repeat count]
[sweep minimum maximum size-increment]
```

Following is an example of how a loop operates if you use the following keywords and arguments on the **ping mpls** command:

```
Device# ping mpls
 ipv4
 10.131.159.251/32 destination 127.0.0.1 127.0.0.1 0.0.0.1 repeat 2
sweep 1450 1475 25
Sending 2, [1450..1500]-byte MPLS Echos to 10.131.159.251/32,
     timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
Destination address 127.0.0.1
!
!
Destination address 127.0.0.1
!
!
Destination address 127.0.0.1
!
!
Destination address 127.0.0.1
!
!
```

An **mpls ping** command is sent for each packet size range for each destination address until the end address is reached. For this example, the loop continues in the same manner until the destination address, 127.0.0.1, is reached. The sequence continues until the number is reached that you specified with the **repeat** *count* keyword and argument. For this example, the repeat count is 2. The MPLS LSP Ping loop sequence is as follows:

```
repeat  = 1
  destination address 1 (address-start
)
    for (size from sweep
 minimum
 to maximum
, counting by size-increment
)
      send an lsp ping
  destination address 2 (address-start
 +
address-
increment
)
    for (size from sweep
 minimum
 to maximum
, counting by size-increment
)
      send an lsp ping
```

```
  destination address 3 (address-start
 +
address-
increment
 +
address-
increment
)
     for (size from sweep
 minimum
 to maximum
, counting by size-increment
)
      send an lsp ping
.
.
.
  until destination address = address-end
.
.
.
until repeat = count
```

Possible Loop with MPLS LSP Traceroute

With the MPLS LSP Traceroute feature, loops can occur if you use the User Datagram Protocol (UDP) destination address range option and the time-to-live option.

```
trace mpls
 {ipv4

destination-address destination-mask
 [destination

address-start
address-end

address-increment
] | traffic-eng
 tunnel-interface

tunnel-number
[ttl
 maximum-
time-to-live
]
```

Here is an example of how a loop operates if you use the following keywords and arguments on the **trace mpls** command:

```
Device# trace mpls
ipv4
 10.131.159.251/32 destination 127.0.0.1 127.0.0.1 1 ttl 5
Tracing MPLS Label Switched Path to 10.131.159.251/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
Destination address 127.0.0.1
  0 10.131.191.230 MRU 1500 [Labels: 19 Exp: 0]
R 1 10.131.159.226 MRU 1504 [implicit-null] 40 ms
! 2 10.131.159.225 40 ms
```

```
Destination address 127.0.0.2
  0 10.131.191.230 MRU 1500 [Labels: 19 Exp: 0]
R 1 10.131.159.226 MRU 1504 [implicit-null] 40 ms
! 2 10.131.159.225 40 ms
Destination address 127.0.0.3
  0 10.131.191.230 MRU 1500 [Labels: 19 Exp: 0]
R 1 10.131.159.226 MRU 1504 [implicit-null] 40 ms
! 2 10.131.159.225 48 ms
```

An **mpls trace** command is sent for each TTL from 1 to the maximum TTL (**ttl** *maximum-time-to-live* keyword and argument) for each destination address until the address specified with the destination *end-address* argument is reached. For this example, the maximum TTL is 5 and the end destination address is 127.0.0.1. The MPLS LSP Traceroute loop sequence is as follows:

```
destination address 1 (address-start
)
  for (ttl
 from 1 to maximum-time-to-live
)
    send an lsp trace
destination address 2 (address-start
 + address-increment
)
  for (ttl
 from 1 to maximum-time-to-live
)
    send an lsp trace
destination address 3 (address-start
 + address-increment
 + address-increment
)
  for (ttl
 from 1 to
maximum-time-to-live)
    send an lsp trace
.
.
.
until destination address = address-end
```

MPLS Echo Request Packets Not Forwarded by IP

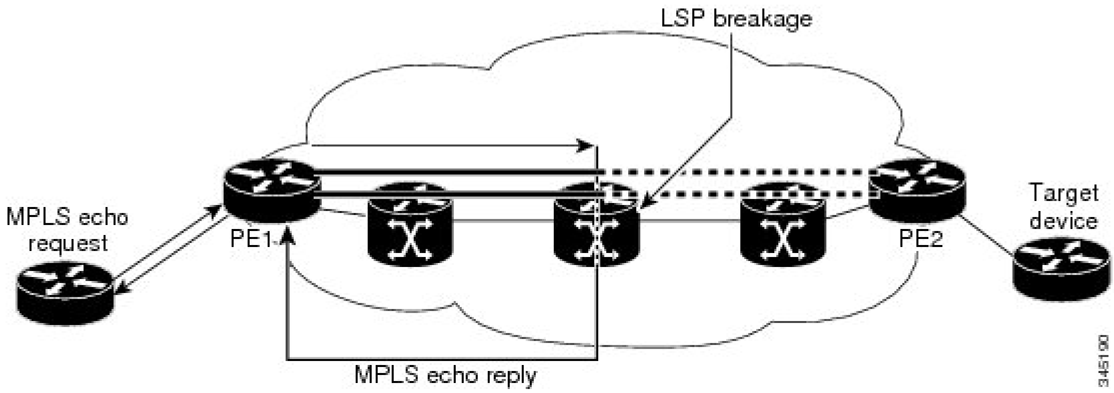
Multiprotocol Label Switching (MPLS) echo request packets sent during a label switched path (LSP) ping are never forwarded by IP. The IP header destination address field in an MPLS echo request packet is a 127.*x.y.z* /8 address. Devices should not forward packets using a 127.*x.y.z* /8 address. The 127.*x.y.z* /8 address corresponds to an address for the local host.

The use of a 127.*x* .*y* .*z* address as a destination address of the User Datagram Protocol (UDP) packet is significant in that the MPLS echo request packet fails to make it to the target device if a transit device does not label switch the LSP. This allows for the detection of LSP breakages.

- If an LSP breakage occurs at a transit device, the MPLS echo packet is not forwarded, but consumed by the device.
- If the LSP is intact, the MPLS echo packet reaches the target device and is processed by the terminal point of the LSP.

The figure below shows the path of the MPLS echo request and reply when a transit device fails to label switch a packet in an LSP.

Figure 9: Path When Transit Device Fails to Label Switch a Packet



Note An Any Transport over MPLS (AToM) payload does not contain usable forwarding information at a transit device because the payload might not be an IP packet. An MPLS virtual private network (VPN) packet, although an IP packet, does not contain usable forwarding information at a transit device because the destination IP address is only significant to the virtual routing and forwarding (VRF) instances at the endpoints of the MPLS network.

Information Provided by the Device Processing LSP Ping or LSP Traceroute

The table below describes the characters that the device processing an LSP ping or LSP traceroute packet returns to the sender about the failure or success of the request.

You can also view the return code for an MPLS LSP Ping operation if you enter the **ping mpls verbose** command.

Table 13: LSP Ping and Traceroute Reply Characters

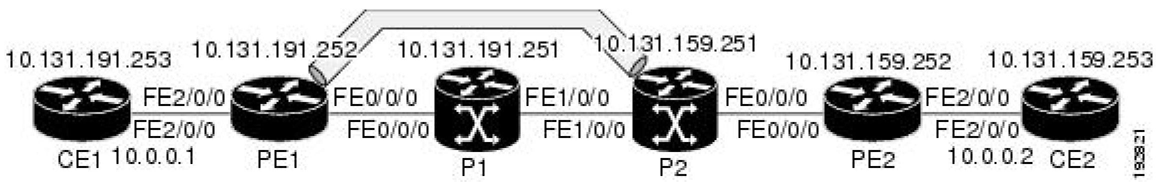
Character	Meaning
Period "."	A timeout occurs before the target device can reply.
U	The target device is unreachable.
R	The device processing the Multiprotocol Label Switching (MPLS) echo request is a downstream device but is not the destination.
Exclamation mark "!"	Replying device is an egress for the destination.
Q	Echo request was not successfully transmitted. This could be returned because of insufficient memory or more probably because no label switched path (LSP) exists that matches the Forwarding Equivalence Class (FEC) information.
C	Replying device rejected the echo request because it was malformed.

MTU Discovery in an LSP

During an MPLS LSP Ping, Multiprotocol Label Switching (MPLS) echo request packets are sent with the IP packet attribute set to do not fragment. That is, the DF bit is set in the IP header of the packet. This allows you to use the MPLS echo request to test for the MTU that can be supported for the packet through the label switched path (LSP) without fragmentation.

The figure below shows a sample network with a single LSP from PE1 to PE2 formed with labels advertised by means of LDP.

Figure 10: Sample Network with LSP—Labels Advertised by LDP



You can determine the maximum receive unit (MRU) at each hop by tracing the LSP using the MPLS Traceroute feature. The MRU is the maximum size of a labeled packet that can be forwarded through an LSP. The following example shows the results of a **trace mpls** command when the LSP is formed with labels created by the Label Distribution Protocol (LDP):

```
Device# trace mpls ipv4 10.131.159.252/32
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
  0 10.131.191.230 MRU 1496 [Labels: 22/19 Exp: 0/0]
R 1 10.131.159.226 MRU 1500 [Labels: 19 Exp: 0] 40 ms
R 2 10.131.159.229 MRU 1504 [implicit-null] 28 ms
! 3 10.131.159.230 40 ms
```

You can determine the MRU for the LSP at each hop through the use of the **show forwarding detail** command:

```
Device# show mpls forwarding 10.131.159.252 detail

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
22     19          10.131.159.252/32 0          Tu1        point2point
        MAC/Encaps=14/22, MRU=1496, Tag Stack{22 19}, via Et0/0
        AABBCC009700AABBCC0098008847 0001600000013000
        No output feature configured
```

To determine the maximum sized echo request that will fit on the LSP, you can find the IP MTU by using the **show interface** *type number* command.

```
Device# show interface e0/0

FastEthernet0/0/0 is up, line protocol is up
  Hardware is Lance, address is aabb.cc00.9800 (bia aabb.cc00.9800)
  Internet address is 10.131.191.230/30
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load ½55
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:00:01, output hang never
```

```
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     377795 packets input, 33969220 bytes, 0 no buffer
     Received 231137 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     441772 packets output, 40401350 bytes, 0 underruns
     0 output errors, 0 collisions, 10 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

The IP MTU in the **show interface** *type number* example is 1500 bytes. Subtract the number of bytes corresponding to the label stack from the MTU number. From the output of the **show mpls forwarding** command, the Tag stack consists of one label (21). Therefore, the largest MPLS echo request packet that can be sent in the LSP, shown in the figure above, is 1500 - (2 x 4) = 1492.

You can validate this by using the following **ping mpls** command:

```
Device# ping mpls ipv4 10.131.159.252/32 sweep 1492 1500 1 repeat 1
Sending 1, [1492..1500]-byte MPLS Echos to 10.131.159.252/32,
     timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
!QQQQQQQQ
Success rate is 11 percent (1/9), round-trip min/avg/max = 40/40/40 ms
```

In this command, only packets of 1492 bytes are sent successfully, as indicated by the exclamation point (!). Packets of byte sizes 1493 to 1500 are source-quenched, as indicated by the Q.

You can pad an MPLS echo request so that a payload of a given size can be tested. The pad TLV is useful when you use the MPLS echo request to discover the MTU supportable by an LSP. MTU discovery is extremely important for applications like AToM that contain non-IP payloads that cannot be fragmented.

LSP Network Management

To manage a Multiprotocol Label Switching (MPLS) network you must have the ability to monitor label switched paths (LSPs) and quickly isolate MPLS forwarding problems. You need ways to characterize the liveliness of an LSP and reliably detect when a label switched path fails to deliver user traffic.

You can use MPLS LSP Ping to verify the LSP that is used to transport packets destined for IPv4 Label Distribution Protocol (LDP) prefixes, traffic engineering (TE) tunnels, and Any Transport over MPLS pseudowire Forwarding Equivalence Classes (AToM PW FECs). You can use MPLS LSP Traceroute to trace LSPs that are used to carry packets destined for IPv4 LDP prefixes and TE tunnel FECs.

An MPLS echo request is sent through an LSP to validate it. A TTL expiration or LSP breakage causes the transit device to process the echo request before it gets to the intended destination and returns an MPLS echo reply that contains an explanatory reply code to the originator of the echo request.

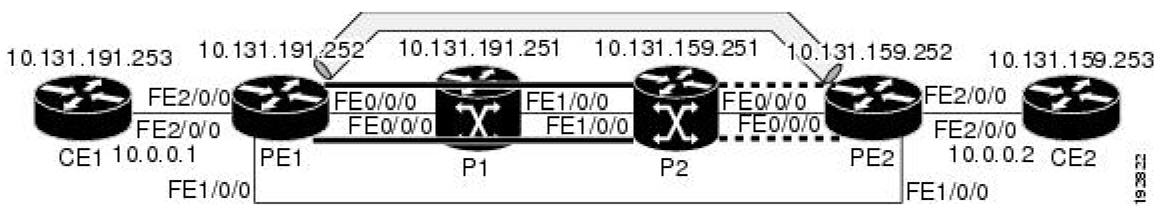
The successful echo request is processed at the egress of the LSP. The echo reply is sent via an IP path, an MPLS path, or a combination of both back to the originator of the echo request.

ICMP ping and trace Commands and Troubleshooting

Internet Control Message Protocol (ICMP) **ping** and **trace** commands are often used to help diagnose the root cause of a failure. When a label switched path (LSP) is broken, the packet might make its way to the target device by way of IP forwarding, thus making ICMP ping and traceroute unreliable for detecting Multiprotocol Label Switching (MPLS) forwarding problems. The MPLS LSP Ping, Traceroute and AToM VCCV feature extends this diagnostic and troubleshooting ability to the MPLS network and handles inconsistencies between the IP and MPLS forwarding tables, inconsistencies in the MPLS control and data plane, and problems with the reply path.

The figure below shows a sample topology with a Label Distribution Protocol (LDP) LSP and traffic engineering (TE) tunnel LSP.

Figure 11: Sample Topology with LDP and TE Tunnel LSPs



This section contains the following topics:

MPLS LSP Ping and Traceroute Discovers LSP Breakage

Configuration for Sample Topology

These are sample topology configurations for the troubleshooting examples in the following sections (see the figure above). There are the six sample device configurations.

Device CE1 Configuration

```
version 12.0
!
hostname ce1
!
enable password lab
!
interface Loopback0
 ip address 10.131.191.253 255.255.255.255
 no ip directed-broadcast
!
interface GigabitEthernet0/0/0
 ip address 10.0.0.1 255.255.255.255
 no ip directed-broadcast
 no keepalive
 no cdp enable
!
end
```

Device PE1 Configuration

```
version 12.0
!
hostname pe1
```

```
!
ip cef
mpls label protocol ldp
mpls traffic-eng tunnels
no mpls traffic-eng auto-bw timers frequency 0
mpls ldp discovery targeted-hello accept
!
interface Loopback0
 ip address 10.131.191.252 255.255.255.255
 no ip directed-broadcast
!
interface Tunnel1
 ip unnumbered Loopback0
 no ip directed-broadcast
 mpls label protocol ldp
 mpls ip
 tunnel destination 10.131.159.255
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 2 2
 tunnel mpls traffic-eng bandwidth  512
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface Tunnel2
 ip unnumbered Loopback0
 no ip directed-broadcast
 shutdown
 mpls label protocol ldp
 mpls ip
 tunnel destination 10.131.159.255
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth  100
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface GigabitEthernet0/0/0
 ip address 10.131.191.230 255.255.255.255
 no ip directed-broadcast
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
interface GigabitEthernet0/0/1
 ip address 10.131.159.246 255.255.255.255
 no ip directed-broadcast
 no shutdown
 mpls ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
interface GigabitEthernet0/0/2
 no ip address
 no ip directed-broadcast
 no cdp enable
 xconnect 10.131.159.252 333 encapsulation mpls
!
interface GigabitEthernet0/0/3
 no ip address
 no ip directed-broadcast
 shutdown
!
router ospf 1
```

```
 log-adjacency-changes
 passive-interface Loopback0
 network 10.131.159.244 0.0.0.3 area 0
 network 10.131.191.228 0.0.0.3 area 0
 network 10.131.191.232 0.0.0.3 area 0
 network 10.131.191.252 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
!
ip classless

end
```

Device P1 Configuration

```
version 12.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname p1
!
enable password lab
!
ip cef
mpls label protocol ldp
mpls ldp logging neighbor-changes
mpls traffic-eng tunnels
no mpls traffic-eng auto-bw timers frequency 0
mpls ldp discovery targeted-hello accept
!
interface Loopback0
 ip address 10.131.191.251 255.255.255.255
 no ip directed-broadcast
!
interface GigabitEthernet0/0/0
 ip address 10.131.191.229 255.255.255.255
 no ip directed-broadcast
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
interface GigabitEthernet0/0/1
 ip address 10.131.159.226 255.255.255.255
 no ip directed-broadcast
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
router ospf 1
 log-adjacency-changes
 passive-interface Loopback0
 network 10.131.159.224 0.0.0.3 area 0
 network 10.131.191.228 0.0.0.3 area 0
 network 10.131.191.251 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
!
end
```

Device P2 Configuration

```
version 12.0
hostname p2
!
ip cef
mpls label protocol ldp
mpls ldp logging neighbor-changes
mpls traffic-eng tunnels
no mpls traffic-eng auto-bw timers frequency 0
mpls ldp discovery directed-hello accept
!
!
interface Loopback0
 ip address 10.131.159.251 255.255.255.255
 no ip directed-broadcast
!
interface GigabitEthernet0/0/0
 ip address 10.131.159.229 255.255.255.255
 no ip directed-broadcast
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
interface GigabitEthernet0/0/1
 ip address 10.131.159.225 255.255.255.255
 no ip directed-broadcast
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
router ospf 1
 log-adjacency-changes
 passive-interface Loopback0
 network 10.131.159.224 0.0.0.3 area 0
 network 10.131.159.228 0.0.0.3 area 0
 network 10.131.159.251 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
!
end
```

Device PE2 Configuration

```
version 12.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname pe2
!
logging snmp-authfail
enable password lab
!
clock timezone EST -5
ip subnet-zero
ip cef
no ip domain-lookup
mpls label protocol ldp
mpls ldp logging neighbor-changes
```

```
mpls ldp explicit-null
mpls traffic-eng tunnels
no mpls traffic-eng auto-bw timers frequency 0
tag-switching tdp discovery directed-hello accept
frame-relay switching
!
!
interface Loopback0
 ip address 10.131.159.252 255.255.255.255
 no ip directed-broadcast
!
interface Tunnel0
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 10.131.191.252
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng path-option 5 explicit name as1pe-long-path
!
interface GigabitEthernet0/0/0
 ip address 10.131.159.230 255.255.255.255
 no ip directed-broadcast
 mpls traffic-eng tunnels
 tag-switching ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
interface GigabitEthernet0/0/1
 ip address 10.131.159.245 255.255.255.255
 no ip directed-broadcast
 mpls traffic-eng tunnels
 tag-switching ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
interface GigabitEthernet0/0/2
 no ip address
 no ip directed-broadcast
 no cdp enable
 xconnect 10.131.191.252 333 encapsulation mpls
!
interface GigabitEthernet0/0/3
 no ip address
 no ip directed-broadcast
!
interface Serial0/0/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Serial0/0/1
 no ip address
 no ip directed-broadcast
 shutdown
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 passive-interface Loopback0
 network 10.131.122.0 0.0.0.3 area 0
 network 10.131.159.228 0.0.0.3 area 0
 network 10.131.159.232 0.0.0.3 area 0
 network 10.131.159.244 0.0.0.3 area 0
 network 10.131.159.252 0.0.0.0 area 0
```

```
!
ip classless
!
!
ip explicit-path name as1pe-long-path enable
 next-address 10.131.159.229
 next-address 10.131.159.226
 next-address 10.131.191.230
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 exec-timeout 0 0
 password lab
 login
!
end
```

Device CE2 Configuration

```
version 12.0
!
hostname ce2
!
enable password lab
!
interface Loopback0
 ip address 10.131.159.253 255.255.255.255
 no ip directed-broadcast
!
interface GigabitEthernet0/0/2
 ip address 10.0.0.2 255.255.255.255
 no ip directed-broadcast
 no keepalive
 no cdp enable
!
end
```

Verifying That the LSP Is Set Up Correctly

A **show mpls forwarding-table** command shows that tunnel 1 is in the Multiprotocol Label Switching (MPLS) forwarding table.

```
Device# show mpls forwarding-table 10.131.159.252

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
22     19
     [T] 10.131.159.252/32 0           Tu1
        point2point
[T]     Forwarding through a TSP tunnel.
        View additional tagging info with the 'detail' option
```

A **show mpls traffic-eng tunnels tunnel 1** command entered at PE1 displays information about tunnel 1 and verifies that it is forwarding packets with an out label of 22.

```
Device# show mpls traffic-eng tunnels tunnel 1

Name: PE1_t1                           (Tunnel1) Destination: 10.131.159.251
```

```
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type dynamic (Basis for Setup, path weight 20)
  Config Parameters:
    Bandwidth: 512      kbps (Global)  Priority: 2  2   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 512      bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: dynamic path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  InLabel  :  -
  OutLabel : FastEthernet0/0/0, 22
  RSVP Signalling Info:
       Src 10.131.191.252, Dst 10.131.159.251, Tun_Id 1, Tun_Instance 28
    RSVP Path Info:
      My Address: 10.131.191.230
      Explicit Route: 10.131.191.229 10.131.159.226 10.131.159.225 10.131.159.251
      Record   Route:   NONE
      Tspec: ave rate=512 kbits, burst=1000 bytes, peak rate=512 kbits
    RSVP Resv Info:
      Record   Route:   NONE
      Fspec: ave rate=512 kbits, burst=1000 bytes, peak rate=512 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 20 (TE)
    Explicit Route: 10.131.191.230 10.131.191.229 10.131.159.226 10.131.159.225
                    10.131.159.251
  History:
    Tunnel:
      Time since created: 9 days, 14 hours, 12 minutes
      Time since path change: 2 minutes, 18 seconds
    Current LSP:
      Uptime: 2 minutes, 18 seconds
    Prior LSP:
      ID: path option 1 [3]
      Removal Trigger: tunnel shutdown
```

A **trace mpls** command issued at PE1 verifies that packets with 22 as the outermost label and 19 as the end of stack label are forwarded from PE1 to PE2.

```
Device# trace mpls ipv4 10.131.159.252/32
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
  0 10.131.191.230 MRU 1496 [Labels: 22/19
 Exp: 0/0]
R 1 10.131.159.226 MRU 1504 [Labels: 19 Exp: 0] 40 ms
R 2 10.131.159.229 MRU 1504 [implicit-null] 28 ms
! 3 10.131.159.230 40 ms
```

The MPLS LSP Traceroute to PE2 is successful, as indicated by the exclamation point (!).

Discovering LSP Breakage

A Label Distribution Protocol (LDP) target-session is established between devices PE1 and P2, as shown in the output of the following **show mpls ldp discovery** command:

```
Device# show mpls ldp discovery

 Local LDP Identifier:
```

```
    10.131.191.252:0
    Discovery Sources:
    Interfaces:
        GigabitEthernet0/0/0 (ldp): xmit/recv
            LDP Id: 10.131.191.251:0
        Tunnel1 (ldp): Targeted -> 10.131.159.251
    Targeted Hellos:
        10.131.191.252 -> 10.131.159.252 (ldp): active/passive, xmit/recv
            LDP Id: 10.131.159.252:0
        10.131.191.252 -> 10.131.159.251 (ldp): active, xmit/recv
            LDP Id: 10.131.159.251:0
```

Enter the following command on the P2 device in global configuration mode:

```
Device# no mpls ldp discovery targeted-hello accept
```

The LDP configuration change causes the targeted LDP session between the headend and tailend of the traffic engineering (TE) tunnel to go down. Labels for IPv4 prefixes learned by P2 are not advertised to PE1. Thus, all IP prefixes reachable by P2 are reachable by PE1 only through IP (not MPLS). In other words, packets destined for those prefixes through Tunnel 1 at PE1 will be IP switched at P2 (which is undesirable).

The following **show mpls ldp discovery** command shows that the LDP targeted-session is down:

```
Device# show mpls ldp discovery

 Local LDP Identifier:
    10.131.191.252:0
    Discovery Sources:
    Interfaces:
       GigabitEthernet0/0/0 (ldp): xmit/recv
            LDP Id: 10.131.191.251:0
        Tunnel1 (ldp): Targeted -> 10.131.159.251
    Targeted Hellos:
        10.131.191.252 -> 10.131.159.252 (ldp): active/passive, xmit/recv
            LDP Id: 10.131.159.252:0
        10.131.191.252 -> 10.131.159.251 (ldp): active, xmit
```

Enter the **show mpls forwarding-table** command at the PE1 device. The display shows that the outgoing packets are untagged as a result of the LDP configuration changes.

```
Device# show mpls forwarding-table 10.131.159.252

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
22     Untagged[T]
 10.131.159.252/32 0          Tu1        point2point
[T]     Forwarding through a TSP tunnel.
        View additional tagging info with the 'detail' option
```

A **ping mpls** command entered at the PE1 device displays the following:

```
Device# ping mpls ipv4 10.131.159.252/32 repeat 1
Sending 1, 100-byte MPLS Echos to 10.131.159.252/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
R
Success rate is 0 percent (0/1)
```

The **ping mpls** command fails. The R indicates that the sender of the Multiprotocol Label Switching (MPLS) echo reply had a routing entry but no MPLS Forwarding Equivalence Class (FEC) . Entering the **ping mpls verbose** command displays the MPLS label switched path (LSP) echo reply sender address and the return code. You should be able to solve the problem by Telneting to the replying device and inspecting its forwarding and label tables. You might need to look at the neighboring upstream device as well, because the breakage might be on the upstream device.

```
Device# ping mpls ipv4 10.131.159.252/32 repeat 1 verbose
Sending 1, 100-byte MPLS Echos to 10.131.159.252/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
R   10.131.159.225, return code 6
Success rate is 0 percent (0/1)
```

Alternatively, use the LSP **traceroute** command to figure out which device caused the breakage. In the following example, for subsequent values of TTL greater than 2, the same device keeps responding (10.131.159.225). This suggests that the MPLS echo request keeps getting processed by the device regardless of the TTL. Inspection of the label stack shows that P1 pops the last label and forwards the packet to P2 as an IP packet. This explains why the packet keeps getting processed by P2. MPLS echo request packets cannot be forwarded by use of the destination address in the IP header because the address is set to a 127/8 address.

```
Device# trace mpls ipv4 10.131.159.252/32 ttl 5
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
  0 10.131.191.230 MRU 1500 [Labels: 22 Exp: 0]
R 1 10.131.159.226 MRU 1504 [implicit-null] 40 ms
R 2 10.131.159.225 40 ms
R 3 10.131.159.225 40 ms
R 4 10.131.159.225 40 ms
R 5 10.131.159.225 40 ms
```

MPLS LSP Traceroute Tracks Untagged Cases

This troubleshooting section contains examples of how to use MPLS LSP Traceroute to determine potential issues with packets that are tagged as implicit null and packets that are untagged.

Untagged output interfaces at a penultimate hop do not impact the forwarding of IP packets through a label switched path (LSP) because the forwarding decision is made at the penultimate hop through use of the incoming label. The untagged case causes Any Transport over Multiprotocol Label Switching (AToM) and MPLS virtual private network (VPN) traffic to be dropped at the penultimate hop.

Troubleshooting Implicit Null Cases

In the following example, Tunnel 1 is shut down, and only a label switched path (LSP) formed with Label Distribution Protocol (LDP) labels is established. An implicit null is advertised between the P2 and PE2 devices. Entering an MPLS LSP Traceroute at the PE1 device results in the following display:

```
Device# trace mpls ipv4 10.131.159.252/32
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
```

```
       'R' - downstream router but not target
Type escape sequence to abort.
  0 10.131.191.230 MRU 1500 [Labels: 20 Exp: 0]
R 1 10.131.159.226 MRU 1500 [Labels: 19 Exp: 0] 80 ms
R 2 10.131.159.229 MRU 1504 [implicit-null] 28 ms
! 3 10.131.159.230 40 ms
```

This output shows that packets are forwarded from P2 to PE2 with an implicit-null label. Address 10.131.159.229 is configured for the P2 Fast Ethernet 0/0/0 out interface for the PE2 device.

Troubleshooting Untagged Cases

Untagged cases are valid configurations for Interior Gateway Protocol (IGP) label switched paths (LSPs) that could cause problems for Multiprotocol Label Switching (MPLS) virtual private networks (VPNs).

A **show mpls forwarding-table** command and a **show mpls ldp discovery** command issued at the P2 device show that the Label Distribution Protocol (LDP) is properly set up:

```
Device# show mpls forwarding-table 10.131.159.252

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
19     Pop tag     10.131.159.252/32 0          Et0/0      10.131.159.230
Device# show mpls ldp discovery
 Local LDP Identifier:
    10.131.159.251:0
    Discovery Sources:
    Interfaces:
          GigabitEthernet0/0/0 (ldp): xmit/recv
            LDP Id: 10.131.159.252:0
        FastEthernet1/0/0 (ldp): xmit/recv
            LDP Id: 10.131.191.251:0
```

The **show mpls ldp discovery** command output shows thatGigabitEthernet0/0/0, which connects PE2 to P2, is sending and receiving packets.

If a **no mpls ip** command is entered on GigabitEthernet0/0/0, this could prevent an LDP session between the P2 and PE2 devices from being established. A **show mpls ldp discovery** command entered on the PE device shows that the MPLS LDP session with the PE2 device is down:

```
Device# show mpls ldp discovery

 Local LDP Identifier:
    10.131.159.251:0
    Discovery Sources:
    Interfaces:
       GigabitEthernet0/0/0 (ldp): xmit
        FastEthernet1/0/0 (ldp): xmit/recv
            LDP Id: 10.131.191.251:0
```

If the MPLS LDP session to PE2 goes down, the LSP to 10.131.159.252 becomes untagged, as shown by the **show mpls forwarding-table** command:

```
Device# show mpls forwarding-table 10.131.159.252

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
19     Untagged
    10.131.159.252/32 864        Et0/0      10.131.159.230
```

Untagged cases would provide an MPLS LSP Traceroute reply with packets tagged with No Label, as shown in the following display:

```
Device# trace mpls ipv4 10.131.159.252/32
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
  0 10.131.191.230 MRU 1500 [Labels: 20 Exp: 0]
R 1 10.131.159.226 MRU 1500 [Labels: 19 Exp: 0] 80 ms
R 2 10.131.159.229 MRU 1504 [No Label] 28 ms
! 3 10.131.159.230 40 ms
```

MPLS LSP Ping and Traceroute Returns a Q

The Q return code always means that the packet could not be transmitted. The problem can be caused by insufficient memory, but it probably results because a label switched path (LSP) could not be found that matches the Forwarding Equivalence Class (FEC), information that was entered on the command line.

The reason that the packet was not forwarded needs to be determined. To do so, look at the Routing Information Base (RIB), the Forwarding Information Base (FIB), the Label Information Base (LIB), and the MPLS Label Forwarding Information Base (LFIB). Lack of an entry for the FEC in any one of these routing/forwarding bases would return a Q.

The table below lists commands that you can use for troubleshooting when the MPLS echo reply returns a Q.

Table 14: Troubleshooting a Q

Database	Command to View Contents
Routing Information Base	**show ip route**
Label Information Base and MPLS Forwarding Information Base	**show mpls forwarding-table detail**

The following example shows a **ping mpls** command where the MPLS echo request is not transmitted, as shown by the returned Qs:

```
Device# ping mpls ipv4 10.0.0.1/32
Sending 5, 100-byte MPLS Echos to 10.0.0.1/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
QQQQQ
Success rate is 0 percent (0/5)
```

A **show mpls forwarding-table** command and a **show ip route** command demonstrate that the address is not in either routing table:

```
Device# show mpls forwarding-table 10.0.0.1

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
Device# show ip route 10.0.0.1

% Subnet not in table
```

The MPLS echo request is not transmitted because the IPv4 address (10.0.0.1) is not found in either the LFIB or the RIB routing table.

Load Balancing for IPv4 LDP LSPs

An Internet Control Message Protocol (ICMP) ping or trace follows one path from the originating device to the target device. Round robin load balancing of IP packets from a source device is used to discover the various output paths to the target IP address.

For MPLS LSP Ping and Traceroute, Cisco devices use the source and destination addresses in the IP header for load balancing when multiple paths exist through the network to a target device. The Cisco implementation of MPLS might check the destination address of an IP payload to accomplish load balancing (this checking depends on the platform).

To check for load balancing paths, you use the 127.*z.y.x* /8 destination address in the **ping mpls ipvr** *ip-address address-mask* **destination** *address-start address-end address-increment* command. The following examples show that different paths are followed to the same destination. This demonstrates that load balancing occurs between the originating device and the target device.

To ensure that the Fast Ethernet interface 1/0/0 on the PE1 device is operational, you enter the following commands on the PE1 device:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface fastethernet 1/0/0
Device(config-if)# no shutdown
Device(config-if)# end
*Dec 31 19:14:10.034: %LINK-3-UPDOWN: Interface FastEthernet1/0/0, changed state to up
*Dec 31 19:14:11.054: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0/0,
changed state to upend
PE1#
*Dec 31 19:14:12.574: %SYS-5-CONFIG_I: Configured from console by console
*Dec 31 19:14:19.334: %OSPF-5-ADJCHG: Process 1, Nbr 10.131.159.252 on FastEthernet1/0/0
from LOADING to FULL, Loading Done
PE1#
```

The following **show mpls forwarding-table** command displays the possible outgoing interfaces and next hops for the prefix 10.131.159.251/32:

```
Device# show mpls forwarding-table 10.131.159.251

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
21     19          10.131.159.251/32 0          FE0/0/0 10.131.191.229
       20          10.131.159.251/32 0          FE1/0/0 10.131.159.245
```

The following **ping mpls** command to 10.131.159.251/32 with a destination UDP address of 127.0.0.1 shows that the path selected has a path index of 0:

```
Device# ping mpls ipv4
 10.131.159.251/32 destination
 127.0.0.1 repeat 1
Sending 1, 100-byte MPLS Echos to 10.131.159.251/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
```

```
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 40/40/40 ms
PE1#
*Dec 29 20:42:40.638: LSPV: Echo Request sent on IPV4 LSP, load_index 2,
pathindex 0
, size 100
*Dec 29 20:42:40.638: 46 00 00 64 00 00 40 00 FF 11 9D 03 0A 83 BF FC
*Dec 29 20:42:40.638: 7F 00 00 01 94 04 00 00 0D AF 0D AF 00 4C 14 70
*Dec 29 20:42:40.638: 00 01 00 00 01 02 00 00 1A 00 00 1C 00 00 00 01
*Dec 29 20:42:40.638: C3 9B 10 40 A3 6C 08 D4 00 00 00 00 00 00 00 00
*Dec 29 20:42:40.638: 00 01 00 09 00 01 00 05 0A 83 9F FB 20 00 03 00
*Dec 29 20:42:40.638: 13 01 AB CD AB CD AB CD AB CD AB CD AB CD AB CD
*Dec 29 20:42:40.638: AB CD AB CD
*Dec 29 20:42:40.678: LSPV: Echo packet received: src 10.131.159.225,
dst 10.131.191.252, size 74
*Dec 29 20:42:40.678: AA BB CC 00 98 01 AA BB CC 00 FC 01 08 00 45 C0
*Dec 29 20:42:40.678: 00 3C 32 D6 00 00 FD 11 15 37 0A 83 9F E1 0A 83
*Dec 29 20:42:40.678: BF FC 0D AF 0D AF 00 28 D1 85 00 01 00 00 02 02
*Dec 29 20:42:40.678: 03 00 1A 00 00 1C 00 00 00 01 C3 9B 10 40 A3 6C
*Dec 29 20:42:40.678: 08 D4 C3 9B 10 40 66 F5 C3 C8
```

The following **ping mpls** command to 10.131.159.251/32 with a destination UDP address of 127.0.0.1 shows that the path selected has a path index of 1:

```
Device# ping mpls ipv4 10.131.159.251/32 dest 127.0.0.1 repeat 1
Sending 1, 100-byte MPLS Echos to 10.131.159.251/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 40/40/40 ms
*Dec 29 20:43:09.518: LSPV: Echo Request sent on IPV4 LSP, load_index 13,
pathindex 1
, size 100
*Dec 29 20:43:09.518: 46 00 00 64 00 00 40 00 FF 11 9D 01 0A 83 BF FC
*Dec 29 20:43:09.518: 7F 00 00 03 94 04 00 00 0D AF 0D AF 00 4C 88 58
*Dec 29 20:43:09.518: 00 01 00 00 01 02 00 00 38 00 00 1D 00 00 00 01
*Dec 29 20:43:09.518: C3 9B 10 5D 84 B3 95 84 00 00 00 00 00 00 00 00
*Dec 29 20:43:09.518: 00 01 00 09 00 01 00 05 0A 83 9F FB 20 00 03 00
*Dec 29 20:43:09.518: 13 01 AB CD AB CD AB CD AB CD AB CD AB CD AB CD
*Dec 29 20:43:09.518: AB CD AB CD
*Dec 29 20:43:09.558: LSPV: Echo packet received: src 10.131.159.229,
dst 10.131.191.252, size 74
*Dec 29 20:43:09.558: AA BB CC 00 98 01 AA BB CC 00 FC 01 08 00 45 C0
*Dec 29 20:43:09.558: 00 3C 32 E9 00 00 FD 11 15 20 0A 83 9F E5 0A 83
*Dec 29 20:43:09.558: BF FC 0D AF 0D AF 00 28 D7 57 00 01 00 00 02 02
*Dec 29 20:43:09.558: 03 00 38 00 00 1D 00 00 00 01 C3 9B 10 5D 84 B3
*Dec 29 20:43:09.558: 95 84 C3 9B 10 5D 48 3D 50 78
```

To see the actual path chosen, you use the **debug mpls lspv packet data** command.

Note The hashing algorithm is nondeterministic. Therefore, the selection of the *address-start* , *address-end* , and *address-increment* arguments for the **destination** keyword might not provide the expected results.

CHAPTER 5

NSR LDP Support

The NSR LDP Support feature allows the Label Distribution Protocol (LDP) to continue to operate across a Router Processor (RP) failure in redundant systems, without losing peer sessions. Before the introduction of nonstop routing (NSR), LDP sessions with peers reset if an RP failover (in a redundant system) or a Cisco In-Service Software Upgrade (ISSU) occurred. When peers reset, traffic is lost while the session is down. Protocol reconvergence occurs after the session is reestablished.

When NSR is enabled, RP failover and Cisco ISSU events are not visible to the peer device, and the LDP sessions that were established prior to failover do not flap. The protocol state learned from the peers persists across an RP failover or Cisco ISSU event and does not need to be relearned.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for NSR LDP Support

The Label Distribution Protocol (LDP) must be up and running on the standby Route Processor (RP) for NSR LDP Support to work.

Information About NSR LDP Support

Roles of the Standby Route Processor and Standby LDP

For the NSR LDP Support feature to work, the Label Distribution Protocol (LDP) must be up and running on the standby Route Processor (RP). The LDP component running on the active RP is called the active LDP, and the LDP component running on the standby RP is called the standby LDP.

When nonstop routing (NSR) is enabled, the standby LDP runs independently from the active LDP, but with the assistance of some software components. The standby LDP maintains LDP session states and database information, ready to take over for the active LDP if the failover occurs.

Standby LDP maintains its local database by querying or receiving notifications of interface status change, configuration changes from the CLI, and checkpoints from the active LDP for other information that is not directly available on the standby RP.

To keep the protocol and session-state information synchronized with the active LDP, the standby LDP depends on TCP to replicate all LDP messages on the active RP to the standby RP. The standby LDP processes all received messages, updates its state, but does not send any responses to its neighbors.

The standby LDP performs the following tasks:

- Processes LDP configuration on startup and during steady state
- Processes active LDP checkpoints of state and session information such as LDP adjacencies, remote addresses, remote bindings, and so forth
- Builds its database of local interfaces
- Processes interface change events
- Receives and processes all LDP messages replicated by TCP
- Updates remote address and label databases

After a switchover and notification that the RP has become active, the standby LDP takes over the role of the active LDP and performs the following tasks:

- Sends hello messages immediately to prevent neighbors from reaching the discovery timeout and bringing down the session
- Retransmits any protocol-level response that has not been sent by the previous active LDP
- Readvertises label bindings
- Refreshes all forwarding entries
- Processes and responds to any LDP message from its neighbor

When the NSR LDP Support feature is disabled, the active LDP performs the following tasks:

- Stops checkpointing to the standby LDP
- Continues to manage all existing sessions

The standby LDP performs the following tasks:

- Cleans up all session-state information
- Reverses to the behavior before NSR is enabled

LDP Operating States

When the NSR LDP Support feature is enabled, the Label Distribution Protocol (LDP) operates in the following states:

Initial State

In the initial state, the active Label Distribution Protocol (LDP) process sets up the standby LDP to be ready to support nonstop routing (NSR). The active LDP performs the following tasks:

- Replicates all TCP sessions used by LDP with the standby LDP
- Synchronizes all existing session-state information with the standby LDP
- Synchronizes the LDP database with the standby LDP

LDP could be in the initial state because of one of these conditions:

- NSR is enabled
- NSR was enabled and the standby Route Processor (RP) starts up (asymmetric startup)
- System boots up and NSR is configured (symmetric startup)

Steady State

In the steady state, the active and standby Label Distribution Protocol (LDP) databases are synchronized. The active and standby LDP process the same LDP messages and update their states independently. The standby LDP is ready to take over the active LDP role in a switchover event.

On the active Route Processor (RP), the active LDP performs the following tasks:

- Continues to manage all existing sessions and checkpoints any significant session event to the standby LDP (such as adjacency delete, session shutdown, timers)
- Notifies the standby LDP of new adjacencies and neighbors

On the standby RP, the standby LDP performs these tasks:

- Processes all received messages but does not send any messages to its neighbor
- Processes checkpoint information from the active LDP
- Manages session keepalive timers but does not bring down the session if a keepalive timer times out

Post Switchover

In the post switchover state, the standby Label Distribution Protocol (LDP) process takes over the active LDP role while the active Route Processor (RP) is reloading.

Supported NSR Scenarios

The NSR LDP Support feature is supported under the following scenarios:

- Route Processor (RP) failover or node failure

  The Label Distribution Protocol (LDP) keeps the session up during an RP or node failover because the LDP adjacency and session-state information between LDP on the active and standby RPs are synchronized. As sessions are created on the active RP, new adjacencies are synchronized to the standby RP. If a standby RP is brought online after sessions are already up (asymmetric startup), LDP synchronizes the existing session-state information from the active to the standby RP.

- Cisco In-Service Software Upgrade (ISSU)

  LDP supports Cisco ISSU negotiation between RPs when a standby RP comes online for the MPLS LDP IGP Synchronization feature. Current Cisco ISSU negotiation is not impacted by NSR. For NSR, LDP negotiates messages specific to NSR, which are checkpointed during initial synchronization (adjacency and session-state information).

How to Configure NSR LDP Support

Enabling NSR LDP Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ldp nsr**
4. **exit**
5. **show mpls ldp nsr**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls ldp nsr** **Example:** `Device(config)# mpls ldp nsr`	Enables nonstop routing (NSR) for all Label Distribution Protocol (LDP) sessions for both link and targeted.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show mpls ldp nsr** **Example:** `Device# show mpls ldp nsr`	Displays whether NSR is enabled.

Troubleshooting Tips for NSR LDP Support

Use the **debug mpls ldp nsr** command to enable the display of Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) nonstop routing (NSR) debugging events for all NSR sessions or for the specified peer.

Configuration Examples for NSR LDP Support

Example: NSR LDP Configuration

Device 1 Configured with NSR LDP Support

```
Router# show mpls ldp nsr

LDP Non-Stop Routing is enabled
LDP Non-Stop Routing Sessions:
VRF default:
Peer LDP Ident: 3.3.3.3:0 State: Ready

Router# show mpls ldp nsr statistics

Peer: 3.3.3.3:0
In label Request Records created: 0, freed: 0
In label Withdraw Records created: 0, freed: 0
Local Address Withdraw Set: 0, Cleared: 0
Transmit contexts enqueued: 0, dequeued: 0
Total In label Request Records created: 0, freed: 0
Total In label Withdraw Records created: 0, freed: 0
Total Local Address Withdraw Records created: 0, freed: 0
Label Request Acks:
Number of chkpt msg sent: 0
Number of chkpt msg in queue: 0
Number of chkpt msg in state none: 0
Number of chkpt msg in state send: 0
Number of chkpt msg in state wait: 0
Label Withdraw Acks:
Number of chkpt msg sent: 0
Number of chkpt msg in queue: 0
Number of chkpt msg in state none: 0
Number of chkpt msg in state send: 0
Number of chkpt msg in state wait: 0
```

```
Address Withdraw Acks:
Number of chkpt msg sent: 0
Number of chkpt msg in queue: 0
Number of chkpt msg in state none: 0
Number of chkpt msg in state send: 0
Number of chkpt msg in state wait: 0
Session Sync:
Number of session-sync msg sent: 3
Number of address records created: 1
Number of address records freed: 1
Number of dup-address records created: 1
Number of dup-address records freed: 1
Number of remote binding records created: 1
Number of remote binding records freed: 1
Number of capability records created: 1
Number of capability records freed: 1
Number of addr msg in state none: 0
Number of dup-addr msg in state none: 0
Number of remote binding msg in state none: 0
Number of capability msg in state none: 0
Number of addr msg in state send: 0
Number of dup-addr msg in state send: 0
Number of remote binding msg in state send: 0
Number of capability msg in state send: 0
Number of addr msg in state wait: 0
Number of dup-addr msg in state wait: 0
Number of remote binding msg in state wait: 0
Number of capability msg in state wait: 0
Number of sync-done msg sent: 1

Router# show mpls ldp neighbor

Peer LDP Ident: 3.3.3.3:0; Local LDP Ident 2.2.2.2:0
TCP connection: 3.3.3.3.646 - 5.5.5.5.13395
State: Oper; Msgs sent/rcvd: 222/219; Downstream
Up time: 02:44:11
LDP discovery sources:
Port-channel1, Src IP addr: 10.5.1.1
TenGigabitEthernet0/4/1, Src IP addr: 10.3.1.1
TenGigabitEthernet0/0/1, Src IP addr: 10.4.1.1
Addresses bound to peer LDP Ident:
3.3.3.3 10.5.1.1 10.7.1.1 10.6.1.1
10.8.1.1 10.3.1.1 10.4.1.1
```

Device 2 Configured without NSR LDP Support

```
Router# show mpls ldp nsr

LDP Non-Stop Routing is disabled
```

Additional References for NSR LDP Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference

Related Topic	Document Title
LDP configuration tasks	*MPLS Label Distribution Protocol Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password	http://www.cisco.com/cisco/web/support/index.html

Feature Information for NSR LDP Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 15: Feature Information for NSR LDP Support

Feature Name	Release	Feature Information
NSR LDP Support	IOS XE 3.5	This feature was introduced on the Cisco RSP1 Module in this release.
NSR LDP Support	IOS XE 3.13	This feature was introduced on the Cisco RSP2 Module in this release.
NSR LDP Support	IOS XE 3.16	This feature was introduced on the Cisco RSP3 Module in this release.

CHAPTER 6

Flex LSP Overview

Flex LSP also known as Associated Bidirectional LSPs is the combination of static bidirectional MPLS-TP and dynamic MPLS-TE. Flex LSP provides bidirectional label switched paths (LSPs) set up dynamically through Resource Reservation Protocol–Traffic Engineering (RSVP-TE). It does not support non-co routed LSPs.

Flex Label Switched Paths are LSP instances where the forward and the reverse direction paths are setup, monitored and protected independently and associated together during signaling. You use a RSVP Association object to bind the two forward and reverse LSPs together to form either a co-routed or non co-routed associated bidirectional TE tunnel.

You can associate a protecting MPLS-TE tunnel with either a working MPLS-TE LSP, protecting MPLS-TE LSP, or both. The working LSP is the primary LSP backed up by the protecting LSP. When a working LSP goes down, the protecting LSP is automatically activated. You can configure a MPLS-TE tunnel to operate without protection as well.

Effective Cisco IOS XE Release 3.18.1SP, Flex LSP supports inter-area tunnels with non co-routed mode.

Signaling Methods and Object Association for Flex LSPs

This section provides an overview of the association signaling methods for the bidirectional LSPs. Two unidirectional LSPs can be bound to form an associated bidirectional LSP in the following scenarios:

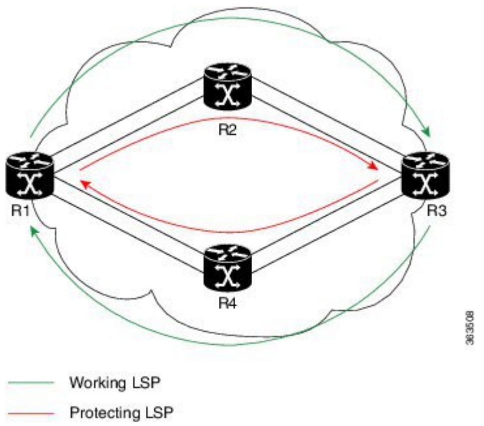
- No unidirectional LSP exists, and both must be established.
- Both unidirectional LSPs exist, but the association must be established.
- One unidirectional LSP exists, but the reverse associated LSP must be established.

Associated Bidirectional Non Co-routed and Co-routed LSPs

This section provides an overview of associated bidirectional non co-routed and co-routed LSPs. Establishment of MPLS TE-LSP involves computation of a path between a head-end node to a tail-end node, signaling along the path, and modification of intermediate nodes along the path. The signaling process ensures bandwidth reservation (if signaled bandwidth is lesser than 0 and programming of forwarding entries).

Path computation is performed by the head-end nodes of both the participating LSPs using Constrained Shortest Path First (CSPF). CSPF is the 'shortest path (measured in terms of cost) that satisfies all relevant LSP TE constraints or attributes, such as required bandwidth, priority and so on.

Associated Bidirectional Non Co-routed LSPs: A non co-routed bidirectional TE LSP follows two different paths, that is, the forward direction LSP path is different than the reverse direction LSP path. Here is an illustration.



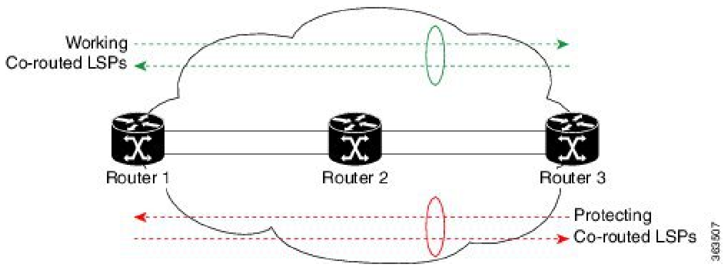
In the above topology:

- The outer paths (in green) are working LSP pairs.
- The inner paths (in red) are protecting LSP pairs.
- Router 1 sets up working LSP to Router 3 and protecting LSP to Router 3 independently.
- Router 3 sets up working LSP to Router 1 and protecting LSP to Router 1 independently.

Non co-routed bidirectional TE LSP is available by default, and no configuration is required.

Note In case of non co-routed LSPs, the head-end nodes relax the constraint on having identical forward and reverse paths. Hence, depending on network state you can have identical forward and reverse paths, though the bidirectional LSP is co-routed.

Associated Bidirectional Co-routed LSPs: A co-routed bidirectional TE LSP denotes a bidirectional tunnel where the forward direction LSP and reverse direction LSP must follow the same path, for example, the same nodes and paths. Here is an illustration.



In the above topology:

- Paths at the top of the figure (in green) indicate working co-routed LSP pairs.
- Paths at the bottom of the figure (in red) indicate protecting co-routed LSP pairs.
- Router 1 sets up working LSP to Router 3 (in red) after performing bidirectional CSPF and sends reverse explicit route object (ERO) to Router 3. Node Router 3 uses the received reverse ERO to set up reverse red working LSP to Router 1.
- Router 3 sets up protecting LSP to Router 1 (in green) after performing bidirectional CSPF and sends reverse ERO to Router 1. Node Router 1 uses the received reverse ERO to set up reverse green protecting LSP to Router 3.

Restrictions for Flex LSP

- Exp-null over Flex-LSP is not supported.
- Flex-LSP does not support tunnel statistics.
- VC (layer 2 VPN ckts) statistics are not supported.
- It is recommended to configure for the following timers for Flex-LSP deployments:

  mpls traffic-eng reoptimize timers frequency 120

  mpls traffic-eng reoptimize timers delay installation 30

  mpls traffic-eng reoptimize timers delay cleanup 90
- The **no mpls ip propagate-tcl** command is not recommended with Flex-LSP. The PREC value of BFD control packet is set to “0”. Therefore, packet prioritization cannot be done at mid-points and BFD flap can occur with traffic congestions.
- It is recommended to configure BFD timers as 10x3 during cable pull testing or in Flex LSP feature deployments.
- 50 msec convergence is not guaranteed for local shut.
- 50 msec convergence is not guaranteed without WRAP protection. WRAP protection is mandatory to achieve 50 msec convergence for remote failures.
- With scale and multiple other feature mix-up, it is possible to see higher convergence.
- TE NSR and IGP NSR are mandatory for RSP switchover.
- Flex LSP is supported with the IPv4 template.
- The **ip rsvp signalling hello** command is not mandatory and it can cause a large punt during the cutover and can lead to unexpected results like BFD flapping.

- VPLS over Flex-LSP is not supported.
- Both IGP and FRR must be configured as clients for single-hop BFD when the WRAP protection is enabled; only FRR cannot be the only client configured at midpoint.
- Layer 3 VPN over Flex-LSP is not supported.
- It is recommended to configure 10x3 BFD timers for cable failures, to achieve 50 msec of convergence.
- Flex LSP is not supported over Port-channel in RSP3.
- It is recommended to configure 10x3 BFD timers for cable failures, to achieve 50 msec of convergence.

Restrictions for Non Co-routed Inter-Area Flex LSP Tunnels

- The dynamic path option feature for TE tunnels (**tunnel mpls traffic-eng path-option number dynamic**) is not supported for inter-area tunnels. An explicit path identifying the area border routers (ABRs) is required.
- The MPLS TE AutoRoute feature (**tunnel mpls traffic-eng autoroute announce**) is not supported for inter-area tunnels.
- Tunnel affinity (**tunnel mpls traffic-eng affinity**) is not supported for inter-area tunnels.
- Tunnel metric (**tunnel mpls traffic-eng path-selection metric**) is not supported for inter-area tunnels.
- BFD is not supported with non co-routed inter-area flex LSP tunnels.

How to Configure Co-routed Flex LSPs

A co-routed bidirectional packet LSP is a combination of two LSPs (one in the forward direction and the other in reverse direction) sharing the same path between a pair of ingress and egress nodes. It is established using the extensions to RSVP-TE. This type of LSP can be used to carry any of the standard types of MPLS-based traffic, including Layer 2 VPNs and Layer 2 circuits. You can configure a single BFD session for the bidirectional LSP (that is, you do not need to configure a BFD session for each LSP in each direction). You can also configure a single standby bidirectional LSP to provide a backup for the primary bidirectional LSP.

The configuration includes the following steps:

1. Enable basic MPLS Traffic Engineering on hostname PE1.
2. Map L2VPN pseudowire to a specific FLEX LSP tunnel.
3. Configure Flex LSP.
4. Enable BFD.
5. Enable Wrap and Fault OAM.
6. Enable BDIs on a core-facing interface.

Configuring Co-routed Flex LSPs

Before you begin

- You must have symmetric source and destination TE router IDs in order for bidirectional LSPs to be associated.
- Tunnels attributes must be configured identically on both sides of co-routed bidirectional LSP.

Note Up to 250 Flex LSP tunnels are supported.

Procedure

```
1. Enable basic MPLS Traffic Engineering on hostname PE1:

mpls traffic-eng tunnels
mpls traffic-eng fault-oam
mpls traffic-eng nsr
router ospf 100
 router-id 1.1.1.1
 nsr
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0

2. Map L2VPN pseudowire to a specific Flex LSP tunnel:

template type pseudowire mpls-te1 (mpls-te1 can be any name)
encapsulation mpls
preferred-path interface Tunnel1 disable-fallback
bandwidth 100

template type pseudowire mpls-te4
encapsulation mpls
preferred-path interface Tunnel4 disable-fallback
bandwidth 100

3. Configure Flex LSP:

interface Tunnel1
bandwidth 1000
ip unnumbered Loopback0
tunnel mode mpls traffic-eng
tunnel destination 22.22.22.22
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 1000
tunnel mpls traffic-eng path-option 1 explicit name BDI1 bandwidth 1000
tunnel mpls traffic-eng path-option protect 1 explicit name BACKUP1 bandwidth 1000
tunnel mpls traffic-eng bidirectional association id 1 source-address 11.11.11.11 global-id
1
NOTE: To bring up the bi-directional tunnels, association ID, source address and global ID
 must match on both sides of the tunnel.
tunnel mpls traffic-eng bidirectional association type co-routed
ip explicit-path name BDI1 enable
next-address 1.11.1.1
next-address 10.1.2.2
next-address 2.22.1.22
```

```
ip explicit-path name BACKUP1 enable
next-address 10.3.11.1.10
next-address 10.4.22.22

4. Enable BFD

bfd-template single-hop BFD_FLEX
interval min-tx 50 min-rx 50 multiplier 3
interface Tunnel1
tunnel mpls traffic-eng bfd encap-mode gal BFD_FLEX

5. Enable Wrap and Fault OAM

interface Tunnel1
tunnel mpls traffic-eng bidirectional association type co-routed fault-oam wrap-protection


6. Enable BDIs on core-facing interface:
NOTE: Since VLANs are not supported, to represent a VLAN interface, BDI must be used towards
 core-facing interface.

interface BDI1
ip address 1.11.1.11 255.255.255.0
ip ospf 1 area 0
mpls traffic-eng tunnels


interface BDI4
ip address 1.11.4.11 255.255.255.0
ip ospf 1 area 0
mpls traffic-eng tunnels

interface GigabitEthernet0/3/1
ip address 10.3.11.11 255.255.255.0
ip ospf 1 area 0
mpls traffic-eng tunnels

interface GigabitEthernet0/3/0
service instance 1 ethernet
encapsulation dot1q 1
rewrite ingress tag pop 1 symmetric
bridge-domain 1
service instance 4 ethernet
encapsulation dot1q 4
rewrite ingress tag pop 1 symmetric
bridge-domain 4
End
```

Verifying the Co-routed Flex LSP Configuration

To verify the FLEX LSP tunnel summary, use the **show mpls traffic-eng tunnels bidirectional-associated concise** command in MPLS tunnel-te interface.

```
Router# show mpls traffic-eng tunnels summary
Signalling Summary:
    LSP Tunnels Process:            running
    Passive LSP Listener:           running
    RSVP Process:                   running
    Forwarding:                     enabled
    auto-tunnel:
    p2p    Disabled (0), id-range:62336-64335

    Periodic reoptimization:        every 3600 seconds, next in 2942 seconds
```

```
    Periodic FRR Promotion:         Not Running
    Periodic auto-bw collection:    every 300 seconds, next in 243 seconds
    SR tunnel max label push:       1 labels
    P2P:
      Head: 100 interfaces,   0 active signalling attempts, 0 established
            87733091 activations,  87733091 deactivations
            144287155 failed activations
            0 SSO recovery attempts, 0 SSO recovered
      Midpoints: 0, Tails: 0

    P2MP:
      Head: 0 interfaces,   0 active signalling attempts, 0 established
            0 sub-LSP activations,  0 sub-LSP deactivations
            0 LSP successful activations,  0 LSP deactivations
            0 SSO recovery attempts, LSP recovered: 0 full, 0 partial, 0 fail
      Midpoints: 0, Tails: 0

Bidirectional Tunnel Summary:
    Tunnel Head: 100 total, 0 connected, 100 associated, 100 co-routed
    LSPs Head:   0 established, 0 proceeding, 0 associated, 0 standby
    LSPs Mid:    0 established, 0 proceeding, 0 associated, 0 standby
    LSPs Tail:   0 established, 0 proceeding, 0 associated, 0 standby
```

To verify the co-routed LSP, use the **Show mpls traffic-eng tunnel bidirectional co-routed** command.

```
Router#Show mpls traffic-eng tunnel bidirectional co-routed

Name: tunnel-te2  Destination: 192.168.0.3
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected
    path option 1,  type dynamic  (Basis for Setup, path weight 3 (reverse 3))
    Bandwidth Requested: 80000 kbps  CT0
Config Parameters:
    Association Type: Single Sided Bidirectional LSPs, Co-routed: Yes
    Association ID: 100, Source: 9.9.9.9[, Global ID: 9]
    Reverse Bandwidth: 2 kbps CT0, Standby: 2 kbps CT0
    BFD Fast Detection: Enabled
    BFD Parameters: Min-interval 10000 ms, Multiplier 3 (default)
     BFD Bringup Timeout: Interval 60 seconds (default)
     BFD Initial Dampening: 16000 ms (default)
     BFD Maximum Dampening: 600000 ms (default)
     BFD Secondary Dampening: 20000 ms (default)
     Periodic LSP Ping: Interval 120 seconds (default)
     BFD Encap Mode: IP (default) | GAL
    Soft Preemption: Enabled, Current Status: Preemption not pending
```

How to Configure Non Co-routed Inter-area Flex LSP Tunnels

Note The working and protect LSPs for PE1 (head-end) is different from PE2 (tail-end).

At PE1 (head-end):

```
interface Tunnel1001
 ip unnumbered Loopback0
 mpls ip
 tunnel mode mpls traffic-eng
 tunnel destination 1.1.1.1
 tunnel mpls traffic-eng priority 7 7
```

```
 tunnel mpls traffic-eng bandwidth 200
 tunnel mpls traffic-eng path-option 1 explicit name ThruHunG verbatim
 tunnel mpls traffic-eng path-option protect 1 explicit name PROT1 verbatim
 tunnel mpls traffic-eng bidirectional association id 1001 source-address 1.1.1.1 global-id
 1001
!
interface Tunnel1002
 ip unnumbered Loopback0
 mpls ip
 tunnel mode mpls traffic-eng
 tunnel destination 1.1.1.1
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 200
 tunnel mpls traffic-eng path-option 1 explicit name ThruHunG verbatim
 tunnel mpls traffic-eng path-option protect 1 explicit name PROT1 verbatim
 tunnel mpls traffic-eng bidirectional association id 1002 source-address 1.1.1.1 global-id
 1002

ip explicit-path name ThruTenG enable
 next-address loose 22.1.1.2
 next-address loose 10.1.1.1
 next-address loose 1.1.1.1
!
ip explicit-path name ThruHunG enable
 next-address loose 23.1.1.2
 next-address loose 10.1.1.1
 next-address loose 1.1.1.1

ip explicit-path name PROT1 enable
 next-address loose 30.1.1.2
 next-address loose 40.1.1.1
 next-address loose 1.1.1.1
```

At PE2 (tail-end):

```
interface Tunnel1001
 ip unnumbered Loopback0
 mpls ip
 tunnel mode mpls traffic-eng
 tunnel destination 4.4.4.4
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 200
 tunnel mpls traffic-eng path-option 1 explicit name ThruTenG verbatim
 tunnel mpls traffic-eng path-option protect 1 explicit name PROT2 verbatim
 tunnel mpls traffic-eng bidirectional association id 1001 source-address 1.1.1.1 global-id
 1001
!
interface Tunnel1002
 ip unnumbered Loopback0
 mpls ip
 tunnel mode mpls traffic-eng
 tunnel destination 4.4.4.4
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 200
 tunnel mpls traffic-eng path-option 1 explicit name ThruTenG verbatim
 tunnel mpls traffic-eng path-option protect 1 explicit name PROT2 verbatim
 tunnel mpls traffic-eng bidirectional association id 1002 source-address 1.1.1.1 global-id
 1002

ip explicit-path name ThruTenG enable
 next-address loose 10.1.1.2
 next-address loose 22.1.1.1
 next-address loose 4.4.4.4
!
ip explicit-path name ThruHunG enable
```

```
 next-address loose 10.1.1.2
 next-address loose 23.1.1.1
 next-address loose 4.4.4.4

ip explicit-path name PROT2 enable
 next-address loose 41.1.1.2
 next-address loose 31.1.1.1
 next-address loose 4.4.4.4
```

Configuring OSFP for Non Co-routed Flex LSP

Note Add the new area into OSPF based on where you want the Inter-area to run.

```
router ospf 1
 router-id 3.3.3.3
 nsr
 nsf cisco
 microloop avoidance
 passive-interface Loopback0
 network 3.3.3.3 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 mpls traffic-eng area 1
```

Verifying the Non Co-routed Inter-area Flex LSP Tunnels

At the PE1

Router# **show mpls traffic-eng tunnels tunnel 1001**

```
Name: PE1_t1001                           (Tunnel1001) Destination: 4.4.4.4
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit (verbatim) ThruTenG (Basis for Setup, path weight 0)
    Path Protection: Requested
    path protect option 1, type explicit (verbatim) PROT2 (Basis for Protect, path weight
0)

  Config Parameters:
    Bandwidth: 200      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute: disabled LockDown: disabled Loadshare: 200 [10000000] bw-based
    auto-bw: disabled
    Association Type: Double Sided Bidirectional LSPs, Co-routed: NO
    Association ID: 1001, Source: 1.1.1.1, Global ID: 1001
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: enabled

  InLabel  :  -
  OutLabel : BDI100, 242
  Next Hop : 10.1.1.2
  Reverse Associated LSP Information:
    Signaled Name: 4.4.4.4 1001
    Tunnel: 1001, Source: 4.4.4.4, Dest: 1.1.1.1, LSP: 9 State: Up
  Lockout Info:
    Locked out: No
```

```
    Lockout Originated By: None
  Association:
    Association Type: Double Sided Bidirectional LSPs
    Association ID: 1001 Source: 1.1.1.1
  Extended Association:
    Global source: 1001
    Extended ID: None
  RSVP Signalling Info:
       Src 1.1.1.1, Dst 4.4.4.4, Tun_Id 1001, Tun_Instance 9
    RSVP Path Info:
      My Address: 10.1.1.1
      Explicit Route: 10.1.1.2 10.1.1.2* 22.1.1.1* 4.4.4.4*
      Record   Route:
      Tspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
    RSVP Resv Info:
      Record   Route:  22.1.1.2 22.1.1.1
      Fspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 2 (TE)
    Explicit Route: 11.1.1.2 20.1.1.1 4.4.4.4
  Reason for the tunnel being down: Bidirectional: standby error from [1.1.1.1][UNK] LSP[8]


  History:
    Tunnel:
      Time since created: 7 minutes, 51 seconds
      Number of LSP IDs (Tun_Instances) used: 9
    Current LSP: [ID: 9]
      Uptime: 5 minutes, 59 seconds
```

At PE2

Router# **show mpls traffic-eng tunnels tunnel 1001**

```
Name: PE2_t1001                           (Tunnel1001) Destination: 1.1.1.1
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit (verbatim) ThruHunG (Basis for Setup, path weight 0)
    Path Protection: Requested
    path protect option 1, type explicit (verbatim) PROT1 (Basis for Protect, path weight
0)

  Config Parameters:
    Bandwidth: 200      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute: disabled LockDown: disabled Loadshare: 200 [10000000] bw-based
    auto-bw: disabled
    Association Type: Double Sided Bidirectional LSPs, Co-routed: NO
    Association ID: 1001, Source: 1.1.1.1, Global ID: 1001
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: enabled

  InLabel  :  -
  OutLabel : BDI221, 980
  Next Hop : 23.1.1.2
  Reverse Associated LSP Information:
    Signaled Name: 1.1.1.1 1001
    Tunnel: 1001, Source: 1.1.1.1, Dest: 4.4.4.4, LSP: 9 State: Up
  Lockout Info:
    Locked out: No
    Lockout Originated By: None
  Association:
    Association Type: Double Sided Bidirectional LSPs
```

```
    Association ID: 1001 Source: 1.1.1.1
  Extended Association:
    Global source: 1001
    Extended ID: None
  RSVP Signalling Info:
       Src 4.4.4.4, Dst 1.1.1.1, Tun_Id 1001, Tun_Instance 9
    RSVP Path Info:
      My Address: 23.1.1.1
      Explicit Route: 23.1.1.2 23.1.1.2* 10.1.1.1* 1.1.1.1*
      Record   Route:
      Tspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
    RSVP Resv Info:
      Record   Route:  10.1.1.2 10.1.1.1
      Fspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 2 (TE)
    Explicit Route: 20.1.1.2 11.1.1.1 1.1.1.1
  Reason for the tunnel being down: Bidirectional: standby error from [4.4.4.4][UNK] LSP[8]


  History:
    Tunnel:
      Time since created: 8 minutes, 9 seconds
      Time since path change: 6 minutes, 10 seconds
      Number of LSP IDs (Tun_Instances) used: 9
    Current LSP: [ID: 9]
      Uptime: 6 minutes, 10 seconds
```

Troubleshooting Flex LSP

Step 1: Verifying that the Flex LSP Tunnel is in UP State

```
Router# show mpls traffic-eng tunnels bidirectional-associated association id 1

P2P TUNNELS/LSPs:
Name: RP1_t3                              (Tunnel3) Destination: 10.5.0.1
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 2, type explicit expl_route_m2_tail (Basis for Setup, path weight 40)
    path option 3, type explicit expl_route_m3_tail
    Path Protection: 0 Common Link(s), 0 Common Node(s)
    path protect option 2, type explicit expl_route_m3_tail (Basis for Protect, path weight
 40)
    path protect option 3, type list name xtd
    Lockout Info:
      Locked Out: No
  Config Parameters:
    Bandwidth: 500      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute: disabled LockDown: disabled Loadshare: 500 [4000000] bw-based
    auto-bw: disabled
    Association Type: Single Sided Bidirectional LSPs, Co-routed: YES
    Association ID: 1, Source: 2.3.4.5, Global ID: 6
    Fault-OAM: disabled
  Active Path Option Parameters:
    State: explicit path option 2 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  InLabel  :  -
  OutLabel : Ethernet0/0, 16
  Next Hop : 10.1.2.2
-------~Full Output not provided ~--------
```

Step 2: Verifying RSVP Signaling

```
Router# show ip rsvp sender detail
PATH:
  Tun Dest:   10.255.255.1  Tun ID: 15  Ext Tun ID: 10.255.255.8
  Tun Sender: 10.255.255.8  LSP ID: 40
  Path refreshes:
    arriving: from PHOP 10.5.2.1 on Et0/1 every 30000 msecs. Timeout in 136 sec
    sent:     to   NHOP 10.1.4.1 on Ethernet0/0
  Session Attr:
    Setup Prio: 7, Holding Prio: 7
    Flags: (0x4) SE Style
    Session Name: R3_t15
  ERO: (incoming)
    10.5.2.2 (Strict IPv4 Prefix, 8 bytes, /32)
    10.1.4.2 (Strict IPv4 Prefix, 8 bytes, /32)
    10.1.4.1 (Strict IPv4 Prefix, 8 bytes, /32)
    10.255.255.1 (Strict IPv4 Prefix, 8 bytes, /32)
  ERO: (outgoing)
    10.1.4.1 (Strict IPv4 Prefix, 8 bytes, /32)
    10.255.255.1 (Strict IPv4 Prefix, 8 bytes, /32)
  ASSOCIATION:
    Extended Association type: Single sided provisioned bidirectional LSPs IPv4
    Association ID: 1, Source: 1.1.1.1
    Global source: 0
    ExtID[0]: 0xAFFFF08
    ExtID[1]: 0x28
-------~Full Output not provided ~--------
```

Step 3: Verifying RSVP Reservation

```
Router# show ip rsvp reservation detail
Reservation:
  Tun Dest:   10.255.255.1  Tun ID: 15  Ext Tun ID: 10.255.255.8
  Tun Sender: 10.255.255.8  LSP ID: 327
  Resv refreshes:
    arriving: from NHOP 10.1.4.1 on Et0/0 every 30000 msecs. Timeout in 382 sec
  Next Hop: 10.1.4.1 on Ethernet0/0
  Label: 23 (outgoing)
  Reservation Style is Shared-Explicit, QoS Service is Controlled-Load
  Resv ID handle: 1200040C.
  Created: 11:08:07 EST Fri Aug 28 2015
  Average Bitrate is 0 bits/sec, Maximum Burst is 1K bytes
  Min Policed Unit: 0 bytes, Max Pkt Size: 1500 bytes
  Status:
  Policy: Accepted. Policy source(s): MPLS/TE
Reservation:
  Tun Dest:   10.255.255.8  Tun ID: 15  Ext Tun ID: 10.255.255.1
  Tun Sender: 10.255.255.1  LSP ID: 338
  Resv refreshes:
    arriving: from NHOP 10.5.2.1 on Et0/1 every 30000 msecs. Timeout in 382 sec
  Next Hop: 10.5.2.1 on Ethernet0/1
  Label: 17 (outgoing)
  Reservation Style is Shared-Explicit, QoS Service is Controlled-Load
  Resv ID handle: 05000410.
  Created: 11:08:07 EST Fri Aug 28 2015
  Average Bitrate is 0 bits/sec, Maximum Burst is 1K bytes
  Min Policed Unit: 0 bytes, Max Pkt Size: 1500 bytes
  RRO:
    10.3.2.2/32, Flags:0x0 (No Local Protection)
    10.3.2.1/32, Flags:0x0 (No Local Protection)
  Status:
  Policy: Accepted. Policy source(s): MPLS/TE
```

Step 4: Verifying Wrap Functionality

```
Router# show mpls traffic-eng tunnels
P2P TUNNELS/LSPs:
Name: R1_t15                              (Tunnel15) Destination: 10.255.255.8
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit Primary (Basis for Setup, path weight 60)
    path option 2, type dynamic
    Path Protection: 0 Common Link(s), 0 Common Node(s)
    path protect option 1, type explicit Secondary (Basis for Protect, path weight 40)
    Lockout Info:
      Locked Out: No
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute: enabled  LockDown: disabled Loadshare: 0 [0] bw-based
    auto-bw: disabled
    Association Type: Single Sided Bidirectional LSPs, Co-routed: YES
    Association ID: 1, Source: 1.1.1.1
    Fault-OAM: enabled, Path-Protection: ready, Wrap-Protection: enabled, Wrap-Capable: Yes

    FlexLSP Event History:
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
```

```
Router# show mpls traffic-eng tunnels protection
P2P TUNNELS:
R1_t15
  LSP Head, Tunnel15, Admin: up, Oper: up
  Src 10.255.255.1, Dest 10.255.255.8, Instance 34
  Fast Reroute Protection: None
  Lockout Info:
    Locked Out: No
  Path Protection: Backup lsp in use.
  Prior Working LSP details:
    LSP ID: 33 (Delayed Clean)
    Deactivates In: (2796) ms
    InLabel  :  -
    OutLabel : Ethernet0/1, 16
    Next Hop : 10.1.4.2
    Reverse Associated LSP Information:
        Signaled Name: 10.255.255.8 15
        Tunnel: 15, Source: 10.255.255.8, Dest: 10.255.255.1, LSP: 29 State: Up
    Lockout Info:
        Locked out: No
        Lockout Originated By: None
    Association:
        Association Type: Single Sided Bidirectional LSPs
        Association ID: 1 Source: 1.1.1.1
--------~Full Output not provided ~--------
```

Step 5: Verifying BFD and OAM Operations

```
Router# show mpls traffic-eng tunnels detail | sec Fault
    Fault-OAM: enabled, Path-Protection: no protection, Wrap-Protection: disabled,
Wrap-Capable: No
    Fault-OAM Events:
      LSP 4638 (deleted) bfd-delete,
          at 07:32:08 IST Fri Jun 3 2016 (1 days, 8 hours, 35 mins, 30 secs ago)
      LSP 4638 (deleted) fault-delete,
          at 07:32:08 IST Fri Jun 3 2016 (1 days, 8 hours, 35 mins, 30 secs ago)
      LSP 4638 (working) bfd-up,
          at 10:15:31 IST Thu Jun 2 2016 (2 days, 5 hours, 52 mins, 7 secs ago)
      LSP 4637 (working) bfd-delete,
          at 10:15:20 IST Thu Jun 2 2016 (2 days, 5 hours, 52 mins, 18 secs ago)
```

```
      LSP 4637 (working) fault-delete,
          at 10:15:20 IST Thu Jun 2 2016 (2 days, 5 hours, 52 mins, 18 secs ago)
      LSP 4636 (working) bfd-delete,
          at 10:15:17 IST Thu Jun 2 2016 (2 days, 5 hours, 52 mins, 21 secs ago)
      LSP 4636 (working) fault-delete,
          at 10:15:17 IST Thu Jun 2 2016 (2 days, 5 hours, 52 mins, 21 secs ago)
-------~Full Output not provided ~-------
```

```
Router# show mpls fault-oam session end-point detail
  MPLS Fault-OAM End-point Sessions
=====================================
    Session handle : 0x6
    Client handle : 0x2B9FAE02B750
    Local label : 18
    Tunnel interface : Tunnel3 (0x15)
    Tunnel number : 3
    LSP number : 49
    Global ID : 0
    Node ID : 10.1.0.1
    Local event : Fault Clear
    Sender Information
        Fault source : End-point
        Refresh seconds : 20
        Initial count : 0
        Fault type : CLR
        Tx Fault-CLR count : 0
        Tx Fault-AIS count : 0
        Tx Fault-LDI count : 0
        Tx Fault-LKR count : 0
        Tx Lockout-CLR count : 0
        Tx Lockout count : 0
        Tx Error count : 0
    Receiver Information
        Source global ID : 0
        Source node ID : 0
        Source intf number : 0
        Fault type : CLR
        Rx Fault-CLR count : 0
        Rx Fault-AIS count : 0
        Rx Fault-LDI count : 0
        Rx Fault-LKR count : 0
        Rx Lockout-CLR count : 0
        Rx Lockout count : 0
        Rx Error count : 0
    -------~Full Output not provided ~-------
```

Step 6: Verifying that Pseudowire is in UP State

```
Router# show mpls l2transport vc vcid 1            (HEAD router)

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
Gi6            Eth VLAN 30                53.0.0.1        1          UP
 #show mpls l2transport vc vcid 1 detail
Local interface: Gi6 up, line protocol up, Eth VLAN 30 up
  Interworking type is Ethernet
  Destination address: 53.0.0.1, VC ID: 1, VC status: up
    Output interface: Tu10, imposed label stack {29 29780}
    Preferred path: Tunnel10,  active
    Required BW = 15000, Admitted BW = 15000
    Default path: ready
    Next hop: point2point
  Create time: 00:01:13, last status change time: 00:01:13
    Last label FSM state change time: 00:01:13
  Signaling protocol: LDP, peer 53.0.0.1:0 up
```

```
Targeted Hello: 52.0.0.1(LDP Id) -> 53.0.0.1, LDP is UP
Graceful restart: configured and enabled
Non stop routing: configured and not enabled

-------~Full Output not provided ~--------
```

Use the **show adjacency tunnel internal** command to view the software forwarding of the tunnel:

```
Router# show adjacency tunnel1 internal | i lsp-num

        GigabitEthernet0/5/2 55.0.0.1 label 21 lsp-num 20
        Path protected by GigabitEthernet0/5/3 label 22 lsp-num 21
        Reopt of working: Null0 0.0.0.0 label none lsp-num 0
        Reopt of protect: Null0 label none lsp-num 0
```

Flex LSP Phase 2

Flex LSP phase 2 is supported effective Cisco IOS XE Everest 16.5.1. Flex LSP also known as Associated Bidirectional Label Switched Paths (LSPs) are LSP instances where the forward and the reverse direction paths are set up, monitored, protected independently, and associated together during signaling. The RSVP Association aims to bind the forward and reverse LSPs together to form either a co-routed or a non co-routed associated bidirectional traffic engineering (TE) tunnel.

The Cisco IOS XE Everest 16.5.1 supports only co-routed Flex LSP tunnels. Flex LSP Phase 2 supports the following features:

- Shared Risk Link Groups (SRLGs) Protection – SRLGs indicate situations where links in a network share a common fiber (or a common physical attribute). If one link fails, the other links in the group may also fail. Links in this group have a shared risk.

  The MPLS-TE SRLG Protection feature enhances backup tunnel path selection so that a backup tunnel avoids using links that are in the same SRLG as interfaces the backup tunnel is protecting.

- Non-revertive

- Sticky

- Hop count and Cost max limit – Each path from the tunnel source node (head-end) to destination node (tail-end) has a number of characteristics, including node hop count and accumulated path cost. Node hop count is the number of nodes along a path excluding the source node.

  To ensure that the ensure the selected path does not exceed the specified hop count, configure a maximum node hop count.

  Each link segment along the path has an associated path cost. Accumulated path cost is the sum of the path cost of all link segments. In the co-routed tunnel case, the accumulated path cost includes both, the forward and reverse link cost. To ensure the selected path does not exceed the specified cost value, configure a cost max limit.

- ECMP min-fill, max-fill – Path calculation selects a path with the lowest accumulated path cost. Sometimes there are multiple paths from the tunnel source node to the destination node, and all these paths may have the same accumulated (and lowest) path cost. These paths are referred to as “Equal Cost Multi Path” (ECMP). In this scenario, path calculation must use other tiebreakers (such as, node hop count and path minimum bandwidth) to select one path. Max-fill is a tiebreaker method that selects from all ECMPs, a path with the least head room, but can still accommodate the path request. This method has the effect of packing the links. Min-fill is a tiebreaker method that selects from all ECMPs, a path with the most head room, and which has a load balance effect over time.

- Restore path option – The restore path option signals a restore LSP after the double failure of both, primary and protect LSPs.

Flex LSP SRLG and Exclude Option for Explicit Path

Use the following commands to configure SRLG on an interface:

```
Router(config)# interface Ethernet0/1
Router(config-if)# srlg gid <1-4294967295>
```

SRLG values configured on MPLS TE enabled interfaces are flooded through IGP (IS-IS or OSPF), and are used by MPLS TE in the following scenarios:

- Restrict protection path to avoid SRLGs of links in the working path.

  ```
  Router(config)# interface Tunnel100
  Router(config-if)# tunnel mpls traffic-eng path-option protect 1 diverse
  srlg [node] lockdown
  ```

- Exclude SRLG in IP Explicit Path.

  ```
  Router(config)# ip explicit-path name EXAMPLE
  Router(cfg-ip-expl-path)# exclude-srlg A.B.C.D
  ```

  The exclude-srlg command specifies an address to get SRLGs from for exclusion.

Note For bidirectional co-routed LSP, both ends of the tunnel must be configured with the same exclude options in explicit path. Otherwise, it is considered a configuration error.

Note Only 32 SRLG values can be configured on an interface. Also, MPLS TE only accepts up to 400 SRLG values configured on a system.

Configuring Flex LSP SRLG and Exclude Option

On the MPLS-TE enabled interface:

```
!
interface GigabitEthernet0/3/0
 srlg gid 10
 srlg gid 20
 srlg gid 30
 srlg gid 40
 srlg gid 50
 ip address 102.103.1.1 255.255.255.0
 ip ospf 1 area 0
 negotiation auto
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 200000 120000 sub-pool 120000
 ip rsvp signalling hello graceful-restart
```

Enable the SRLG on the Flex LSP tunnel:

```
!
interface Tunnel1
```

```
bandwidth 100000
ip unnumbered Loopback0
tunnel mode mpls traffic-eng
tunnel destination 54.4.4.4
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 100000000
tunnel mpls traffic-eng path-option 1 explicit name SRLG
tunnel mpls traffic-eng path-option protect 1 diverse srlg lockdown
tunnel mpls traffic-eng bidirectional association id 1 source-address 52.2.2.2 global-id 1
tunnel mpls traffic-eng bidirectional association type co-routed fault-oam wrap-protection
```

To enable explicit path SRLG:

```
ip explicit-path name SRLG enable
 exclude-srlg 102.103.1.1
```

Verifying the Flex LSP SRLG and Exclude Option

To view the SRLG values in the topology:

```
Router# show mpls traffic-eng topology 52.2.2.2

IGP Id: 52.2.2.2, MPLS TE Id:52.2.2.2 Router Node (ospf 1  area 0) id 16
      link[0]: Broadcast, DR: 102.104.2.2, nbr_node_id:17, gen:54, nbr_p:4640A034
      frag_id: 2, Intf Address: 102.104.2.1
      TE metric: 1, IGP metric: 1, attribute flags: 0x0
      SRLGs: None
      physical_bw: 1000000 (kbps), max_reservable_bw_global: 200000 (kbps)
      max_reservable_bw_sub: 120000 (kbps)

                           Global Pool       Sub Pool
           Total Allocated   Reservable        Reservable
           BW (kbps)         BW (kbps)         BW (kbps)
           ---------------   -----------       ----------
    bw[0]:           0            200000           120000
    bw[1]:           0            200000           120000
    bw[2]:           0            200000           120000
    bw[3]:           0            200000           120000
    bw[4]:           0            200000           120000
    bw[5]:           0            200000           120000
    bw[6]:           0            200000           120000
    bw[7]:           0            200000           120000

      link[1]: Broadcast, DR: 102.103.1.2, nbr_node_id:13, gen:54, nbr_p:4640A36C
      frag_id: 1, Intf Address: 102.103.1.1
      TE metric: 1, IGP metric: 1, attribute flags: 0x0
      SRLGs: 10 20 30 40 50
      physical_bw: 1000000 (kbps), max_reservable_bw_global: 200000 (kbps)
      max_reservable_bw_sub: 120000 (kbps)

Router# show mpls traffic-eng link-management advertisements
Flooding Status:      ready
Configured Areas:     1
IGP Area[1] ID::  ospf 1  area 0
  System Information::
    Flooding Protocol:    OSPF
  Header Information::
    IGP System ID:        52.2.2.2
    MPLS TE Router ID:    52.2.2.2
    Flooded Links:        2
  Link ID::  0 (GigabitEthernet0/3/0)
    Link Subnet Type:     Broadcast
    Link IP Address:      102.103.1.1
    Designated Router:    102.103.1.2
```

```
TE metric:            1
IGP metric:           1
SRLGs:                10 20 30 40 50
Physical Bandwidth:   1000000 kbits/sec
Res. Global BW:       200000 kbits/sec
Res. Sub BW:          120000 kbits/sec
Downstream::
```

Flex LSP Non-Revertive 1:1 Path Protection

Currently, MPLS-TE path protection is triggered every time a fail is detected on a working path (assuming a protect path is configured and available). Once the fail is resolved, TE switches back traffic to the working path. The process of switching back the traffic to the working path is done in a make-before-break fashion but it still does not guarantee that traffic jitter or delay is not introduced (due to different path lengths between working and protect paths). This jitter is not desirable in delay-intolerant applications like Circuit Emulation (CEM).

In Cisco IOS XE Everest 16.5.1, TE no longer switches back to the working path when the working path is restored. Instead, the protected path continues to be the working path. This behavior is achieved by configuring a specific attribute, [non-revertive], to the protecting path-options.

However, if the non-revertive option is configured, TE switches back to the working path.

This is the sequence of actions for MPLS TE when the non-revertive protect path is configured:

1. Detects failure on the current path.
2. Switches to the protect path. The protect path now becomes the active non-revertive path (that is, the path carrying traffic).
3. Signals a new path and marks it as pending. This new pending path must be diverse from the active path, only if the diverse option is configured on the protect path option.
4. Re-optimizes this pending path periodically (but at the same time ensures this is diverse from the active non-revertive protect path).
5. Switches over to the pending path, if the active path fails [traffic is wrapped]. Marks pending path as active path. Re-signals a protect path for the new active path.
6. You can force traffic to a pending path by running the command **mpls traffic-eng switch non-revertive**.

   In this case, the pending path becomes the current active path and the non-revertive path is re-signaled as a protect path for the new current path.

Configuring Flex LSP Non-Revertive Path Protection

The following command is used to configure the non-revertive path protection:

```
[no] tunnel mpls traffic-eng path-option protect [preference] {diverse | explicit {identifier
 [id] | name [name]} | list {identifier [id] | name [name]}} [non-revertive]
```

The following sample configuration illustrates a co-routed Flex LSP enabled with the non-revertive option:

Note Non-revertive must be configured on the both ends of a Flex Tunnel.

```
Router(config)# interface tunnel 1000
Router (config-if)# ip unnumbered loopback0
Router(config-if)# tunnel mode mpls traffic-eng
Router(config-if)# tunnel destination 56.6.6.6
Router(config-if)# tunnel mpls traffic-eng bidir association type co-routed fault-oam wrap
Router(config-if)# tunnel mpls traffic-eng bidir association id 1000 source 55.5.5.5 global
 1000
Router(config-if)# tunnel mpls traffic-eng path-selection hop-limit 5
Router(config-if)# tunnel mpls traffic-eng bandwidth 200
Router(config-if)# tunnel mpls traffic-eng autoroute announce
Router (config-if)# tunnel mpls traffic-eng path-option 10 dynamic
Router(config-if)# tunnel mpls traffic-eng path-option protect 10 diverse non-revertive
lockdown
```

Verifying Flex LSP Non-Revertive Path Protection

```
Router# show mpls traffic-eng tunnels tunnel 1000 detail

Name: Router_t1000                          (Tunnel1000) Destination: 56.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, type dynamic (Basis for Setup, path weight 40)
    Path Protection: 0 Common Link(s), 0 Common Node(s)
    path protect option 10, (LOCKDOWN) (NON-REVERTIVE) type diverse (Basis for Protect,
path weight 80)
    Lockout Info:
      Locked Out: No

  Config Parameters:
    Bandwidth: 200      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    Path-selection Tiebreaker:
      Global: not set   Tunnel Specific: not set   Effective: min-fill (default)
    Hop Limit: 5
    Cost Limit: disabled
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 200 [10000000] bw-based
    auto-bw: disabled
    Protection non-revertive
```

After non-revertive protect path takes over, and primary path is re-signaled, the primary path is in pending reroute stage as below:

```
Router# show mpls traffic-eng tunnels tunnel 2222
Name: RP1_t2222                          (Tunnel2222) Destination: 10.11.0.5
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path protect option 1, type explicit expl_route_m3_m22 (Basis for Protect, path weight
 61)
    path option 1, type dynamic (Basis for Setup, path weight 62)
    path option 2, type explicit expl_route_m4
    Pending Non Revertive is active. Roles switched
    Path Protection: 0 Common Link(s), 0 Common Node(s)
    path protect option 1, type explicit expl_route_m3_m22 (Basis for Protect, path weight
 61)
    path protect option 2, type explicit expl_route_m3_m22
    Lockout Info:
      Locked Out: No

  Config Parameters:
    Bandwidth: 2        kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: disabled LockDown: disabled Loadshare: 2 [1000000000] bw-based
```

```
      auto-bw: disabled
      Protection non-revertive
      Association Type: Single Sided Bidirectional LSPs, Co-routed: YES
      Association ID: 2222, Source: 2.3.4.5, Global ID: 6
      Reverse Bandwidth: 2 kbps, Standby: 2 kbps
     Fault-OAM: enabled, Path-Protection: not ready, Wrap-Protection: disabled, Wrap-Capable:
 No
    Active Path Option Parameters:
      State: explicit path option 1 is active
      BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
    InLabel  :  -
    OutLabel : Ethernet0/2, 17
    Next Hop : 10.2.0.3

  Reverse Associated LSP Information:
      Signaled Name: 10.11.0.5 2222
      Tunnel: 2222, Source: 10.11.0.5, Dest: 10.11.0.1, LSP: 20 State: Up
    Lockout Info:
      Locked out: No
      Lockout Originated By: None
    Association:
      Association Type: Single Sided Bidirectional LSPs
      Association ID: 2222 Source: 2.3.4.5
      Protecting
    Extended Association:
      Global source: 6
      Extended ID:
      0x0A0B0005 (10.11.0.5)
      0x00000014 (0.0.0.20)
      0x00000013 (0.0.0.19)
      0x00000000 (0.0.0.0)
    RSVP Signalling Info:
         Src 10.11.0.1, Dst 10.11.0.5, Tun_Id 2222, Tun_Instance 20
      RSVP Path Info:
        My Address: 10.2.0.1
        Explicit Route: 10.2.0.3 10.99.0.3 10.99.0.22 10.7.0.22
                        10.7.0.5 10.11.0.5
        Record   Route:   NONE
        Tspec: ave rate=2 kbits, burst=1000 bytes, peak rate=2 kbits
      RSVP Resv Info:
        Record   Route:   NONE
        Fspec: ave rate=2 kbits, burst=1000 bytes, peak rate=2 kbits
    Shortest Unconstrained Path Info:
      Path Weight: 60 (TE)
      Explicit Route: 10.2.0.1 10.2.0.3 10.5.0.3 10.5.0.33
                      10.8.0.33 10.8.0.5 10.11.0.5
    History:
      Tunnel:
        Time since created: 1 hours, 1 minutes
        Time since path change: 56 minutes, 11 seconds
        Number of LSP IDs (Tun_Instances) used: 20

   Current LSP: [ID: 20]
        Uptime: 59 minutes, 56 seconds
        Selection: protected failure
      Non Revert LSP: [ID: 33]
        Uptime: 37 seconds
      Prior LSP: [ID: 19]
        ID: path option unknown
        Removal Trigger: path tear

Router# show mpls traffic-eng tunnels tunnel 2222 protection
Router_t2222
  LSP Head, Tunnel2222, Admin: up, Oper: up
  Src 10.11.0.1, Dest 10.11.0.5, Instance 20
```

```
  Fast Reroute Protection: None
  Lockout Info:
    Locked Out: No
  Pending Non Revertive is active. Roles switched
  Path Protection: 0 Common Link(s), 0 Common Node(s)
    Primary lsp path:10.2.0.1 10.2.0.3  Note this is the active non-revertive protect path
                     10.99.0.3 10.99.0.22
                     10.7.0.22 10.7.0.5
                     10.11.0.5
    Protect lsp path:10.3.0.1 10.3.0.4  Note this is the pending path ready to protect
active non-revertive
                     10.6.0.4 10.6.0.44
                     10.9.0.44 10.9.0.5
                     10.11.0.5
    Path Protect Parameters:
      Bandwidth: 2        kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
      Metric Type: TE (default)
    InLabel  :  -
    OutLabel : Ethernet0/3, 19
    Next Hop : 10.3.0.4
    Reverse Associated LSP Information:
        Signaled Name: 10.11.0.5 2222
        Tunnel: 2222, Source: 10.11.0.5, Dest: 10.11.0.1, LSP: 29 State: Up
    Lockout Info:
        Locked out: No
        Lockout Originated By: None
    Association:
        Association Type: Single Sided Bidirectional LSPs
        Association ID: 2222 Source: 2.3.4.5
    Extended Association:
        Global source: 6
        Extended ID:
        0x0A0B0005 (10.11.0.5)
        0x0000001D (0.0.0.29)
        0x00000000 (0.0.0.0)
        0x00000000 (0.0.0.0)
    RSVP Signalling Info:
         Src 10.11.0.1, Dst 10.11.0.5, Tun_Id 2222, Tun_Instance 33
      RSVP Path Info:
        My Address: 10.3.0.1
        Explicit Route: 10.3.0.4 10.6.0.4 10.6.0.44 10.9.0.44
                        10.9.0.5 10.11.0.5
        Record   Route:   NONE
        Tspec: ave rate=2 kbits, burst=1000 bytes, peak rate=2 kbits
      RSVP Resv Info:
        Record   Route:   NONE
        Fspec: ave rate=2 kbits, burst=1000 bytes, peak rate=2 kbits
```

Flex LSP Sticky

Some tunnel configuration changes (for example, bandwidth) or network events (for example, link failure along the path, a higher priority LSP that preempts the current LSP, or a lockout event on a link along the path) can trigger a new LSP to be signaled. In such instances, the current LSP is switched over to the new LSP in a make-before-break manner.

If the tunnel is running on protect LSP make-before-break is not supported.

The new LSP may have a different path, and lockdown does not prevent this path change. This path change can cause packet jitter that may be unacceptable to Circuit Emulation (CEM) services.

To avoid switching over to a new LSP, a new keyword – sticky is introduced in Cisco IOS XE Everest 16.5.1.

```
tunnel mpls traffic-eng path-option 1 dynamic {lockdown {sticky}]
Router(config-if)# tunnel mpls traffic-eng path-option 1 dynamic lockdown ?
  bandwidth  override the bandwidth configured on the tunnel
  sticky     stay on the same path after change in required resources
```

You can configure lockdown as one of LSP attributes, which in turn is extended with the sticky option.

```
Router(config)#mpls traffic-eng lsp attributes EXAMPLE
Router(config-lsp-attr)#lockdown ?
  sticky  stay on the same path after change in required resources
```

Note The sticky option can be configured only on the primary path option.

Once configured, the protect path option with the same path option index is also sticky. This means that either both primary and protect paths are sticky or none of them are sticky.

Use the following command in EXEC mode to manually trigger rerouting the sticky primary and protect path to override the sticky behavior:

```
mpls traffic-eng reroute {Tunnel <num>}
Router# mpls traffic-eng ?
fast-reroute  fast-reroute command
reoptimize    reoptimize traff-eng tunnels
reroute       allow traff-eng tunnels with sticky path to reroute
setup-timer   set up timer expiration
```

Restrictions

- A primary or protect path becomes sticky if the LSP is connected.
- If the path is for Flex LSP tunnel, the LSP must be bi-directionally connected.
- If BFD is configured, the LSP must be BFD connected.

Configuring Flex LSP Sticky Option

Note Both ends of a Flex LSP tunnel must have the same sticky configuration.

```
!
interface Tunnel1000
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 56.6.6.6
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 200
 tunnel mpls traffic-eng path-option 10 dynamic lockdown sticky
 tunnel mpls traffic-eng path-option protect 10 diverse non-revertive lockdown
 tunnel mpls traffic-eng bidirectional association id 1000 source-address 55.5.5.5 global-id
 1000
 tunnel mpls traffic-eng bidirectional association type co-routed fault-oam wrap-protection
end
```

Verifying the Flex LSP Sticky Option

```
Router# show mpls traffic-eng tunnels tunnel 1000 detail

Name: asr167_t1000                        (Tunnel1000) Destination: 54.4.4.4
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (LOCKDOWN & STICKY) type dynamic (Basis for Setup, path weight 4)
    Path Protection: 0 Common Link(s), 0 Common Node(s)
   path protect option 10, (LOCKDOWN & STICKY) type diverse (Basis for Protect, path weight
 8)
    Lockout Info:
      Locked Out: No

  Config Parameters:
    Bandwidth: 200      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    Path-selection Tiebreaker:
      Global: not set   Tunnel Specific: not set   Effective: min-fill (default)
    Hop Limit: disabled
    Cost Limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown(Sticky): enabled  Loadshare: 200 [10000000] bw-based
    auto-bw: disabled
    Association Type: Single Sided Bidirectional LSPs, Co-routed: YES
    Association ID: 1000, Source: 56.6.6.6, Global ID: 1000
    Reverse Bandwidth: 200 kbps, Standby: 200 kbps
   Fault-OAM: enabled, Path-Protection: ready, Wrap-Protection: enabled, Wrap-Capable: Yes

    Fault-OAM Events:
  Active Path Option Parameters:
    State: dynamic path option 10 is active
    BandwidthOverride: disabled  LockDown(Sticky): enabled   Verbatim: disabled
  Binding SID: 28
  Node Hop Count: 2

  InLabel  :  -
  OutLabel : GigabitEthernet0/0/1, 121
  Next Hop : 102.106.12.1
  Reverse Associated LSP Information:
    Signaled Name: 54.4.4.4 1000
    Tunnel: 1000, Source: 54.4.4.4, Dest: 56.6.6.6, LSP: 13 State: Up
  Lockout Info:
    Locked out: No
    Lockout Originated By: None
  Association:
    Association Type: Single Sided Bidirectional LSPs
    Association ID: 1000 Source: 56.6.6.6
  Extended Association:
    Global source: 1000
    Extended ID:
    0x36040404 (54.4.4.4)
    0x0000000D (0.0.0.13)
    0x00000000 (0.0.0.0)
    0x00000000 (0.0.0.0)
  RSVP Signalling Info:
       Src 56.6.6.6, Dst 54.4.4.4, Tun_Id 1000, Tun_Instance 18
    RSVP Path Info:
      My Address: 102.106.12.2
      Explicit Route: 102.106.12.1 102.104.2.1 102.104.2.2 54.4.4.4
      Record   Route:   NONE
      Tspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
    RSVP Resv Info:
      Record   Route:   NONE
```

```
        Fspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
    Shortest Unconstrained Path Info:
      Path Weight: 4 (TE)
      Explicit Route: 102.106.12.2 102.106.12.1 102.104.2.1 102.104.2.2
                      54.4.4.4
    Sticky Primary Path Info:
      Path Option Index: 10
      Explicit Route: 102.106.12.2 102.106.12.1 102.104.2.1 102.104.2.2
                      54.4.4.4

    History:
      Tunnel:
        Time since created: 42 seconds
        Time since path change: 21 seconds
        Number of LSP IDs (Tun_Instances) used: 18
      Current LSP: [ID: 18]
        Uptime: 21 seconds
      Prior LSP: [ID: 16]
        ID: path option unknown
        Removal Trigger: path error

Router# show mpls traffic-eng tunnels tunnel 1000 protection
asr167_t1000
  LSP Head, Tunnel1000, Admin: up, Oper: up
  Src 56.6.6.6, Dest 54.4.4.4, Instance 18
  Fast Reroute Protection: None
  Lockout Info:
    Locked Out: No
  Path Protection: 0 Common Link(s), 0 Common Node(s)
    Primary lsp path: 102.106.12.2 102.106.12.1 102.104.2.1 102.104.2.2
                      54.4.4.4
    Protect lsp path: 101.106.1.2 101.106.1.1 101.105.7.1 101.105.7.2
                      103.105.9.2 103.105.9.1 103.104.10.1 103.104.10.2
                      54.4.4.4
    Path Protect Parameters:
      Bandwidth: 200      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
      Metric Type: TE (default)
    InLabel  :  -
    OutLabel : GigabitEthernet0/0/2, 133
    Next Hop : 101.106.1.1
    Reverse Associated LSP Information:
        Signaled Name: 54.4.4.4 1000
        Tunnel: 1000, Source: 54.4.4.4, Dest: 56.6.6.6, LSP: 15 State: Up
    Lockout Info:
        Locked out: No
        Lockout Originated By: None
    Association:
        Association Type: Single Sided Bidirectional LSPs
        Association ID: 1000 Source: 56.6.6.6
        Protecting
    Extended Association:
        Global source: 1000
        Extended ID:
        0x38060606 (56.6.6.6)
        0x00000013 (0.0.0.19)
        0x00000012 (0.0.0.18)
        0x00000000 (0.0.0.0)
    RSVP Signalling Info:
         Src 56.6.6.6, Dst 54.4.4.4, Tun_Id 1000, Tun_Instance 19
      RSVP Path Info:
        My Address: 101.106.1.2
        Explicit Route: 101.106.1.1 101.105.7.1 101.105.7.2 103.105.9.2
                        103.105.9.1 103.104.10.1 103.104.10.2 54.4.4.4
        Record   Route:   NONE
        Tspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
```

```
    RSVP Resv Info:
      Record   Route:   NONE
      Fspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
Sticky Protect Path Info:
  Path Option Index: 10
  Explicit Route: 101.106.1.2 101.106.1.1 101.105.7.1 101.105.7.2
                  103.105.9.2 103.105.9.1 103.104.10.1 103.104.10.2
                  54.4.4.4
```

Flex LSP Hop Count and Cost-Max Limit

In some scenarios, path hop count is used as an estimate of path delay between a source node and destination node.

The hop-limit is the number of nodes in the path, excluding the headend. For example, a single-hop tunnel has a headend router and a tailend router, but no mid-node routers, a two-hop tunnel has a single mid-node router and atailend router, and so on.

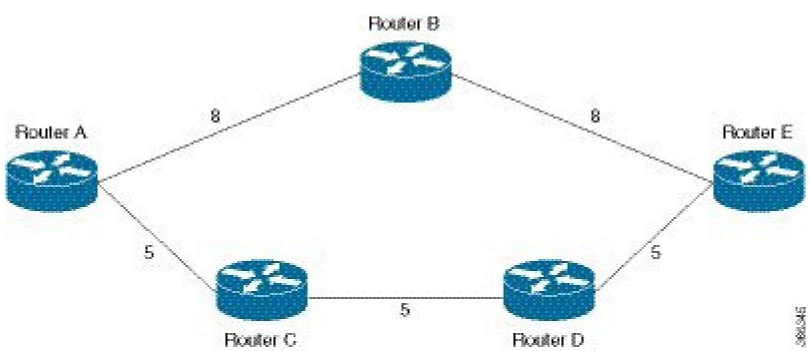
You can configure the hop-limit applicable to a specific tunnel. For example,

```
Router(config)# interface tunnel 1400
Router(config-if)# tunnel mpls traffic-eng path-selection hop-limit <1-255>
```

Consider the following simple topology with only the path cost and path hop count. The number indicated along each link segment is the link cost.

Without any hop-limit, the best path from Router A to Router E as selected by PCALC is [A, C, D, E] since this path has lower accumulated path cost than the other path [A, B, E]. If hop-limit is set to two, then the *best* path that can pass this check is [A, B, E] even though it has higher accumulated path cost.

Figure 12: Topology: Hop Count and Path Cost



Flex LSP Cost-Max Limit

In some scenarios, accumulated path cost is used as an estimate for path delay between a source node and destination node. The intent is to ensure that the accumulated path cost of a chosen path is not higher than a configured limit. A cost limit check is added after the CSPF determines the best path (lowest cost path that satisfied the path request constraints) from source node to destination node, and that for path verification.

Note Ensure that the accumulated path cost for co-routed bidirectional LSP includes both, forward and reverse path cost.

To configure the hop-limit that is applicable to a specific tunnel, use:

```
Router(config)# interface tunnel 1400
Router(config-if)# tunnel mpls traffic-eng path-selection cost-limit <1-4294967295>
```

Configuring Flex LSP Hop Count and Cost-Max Limit

Configure each end of the co-routed bi-directional LSP independently. Ensure that cost-limit and hop-limit on both ends are configured in the same way.

```
Router(config)# interface tunnel 1000
Router(config-if)# ip unnumbered loopback0
Router(config-if)# tunnel mode mpls traffic-eng
Router(config-if)# tunnel destination 56.6.6.6
Router(config-if)# tunnel mpls traffic-eng bidir association type co-routed fault-oam wrap
Router(config-if)# tunnel mpls traffic-eng bidir association id 1000 source 55.5.5.5 global
 1000
Router(config-if)# tunnel mpls traffic-eng path-selection hop-limit 5
Router(config-if)# tunnel mpls traffic-eng bandwidth 200
Router(config-if)# tunnel mpls traffic-eng autoroute announce
Router(config-if)# tunnel mpls traffic-eng path-option 10 dynamic
```

```
Router(config)# interface tunnel 3000
Router(config-if)# ip unnumbered loopback0
Router(config-if)# tunnel mode mpls traffic-eng
Router(config-if)# tunnel destination 56.6.6.6
Router(config-if)# tunnel mpls traffic-eng bidir association type co-routed fault-oam wrap
Router(config-if)# tunnel mpls traffic-eng bidir association id 3000 source 55.5.5.5 global
 3000
Router(config-if)# tunnel mpls traffic-eng path-selection cost-limit 50
Router(config-if)# tunnel mpls traffic-eng bandwidth 200
Router(config-if)# tunnel mpls traffic-eng autoroute announce
Router(config-if)# tunnel mpls traffic-eng path-option 10 dynamic
```

Verifying Flex LSP Hop Count and Cost-Max Limit

```
Router# show mpls traffic-eng tunnels tunnel 1000 detail

Name: Router_t1000                           (Tunnel1000) Destination: 56.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, type dynamic (Basis for Setup, path weight 40)
    Lockout Info:
      Locked Out: No

  Config Parameters:
    Bandwidth: 200      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    Path-selection Tiebreaker:
      Global: not set   Tunnel Specific: not set   Effective: min-fill (default)
    Hop Limit: 5
```

```
Router# show mpls traffic-eng tunnels tunnel 3000 detail

Name: Router_t3000                           (Tunnel3000) Destination: 56.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, type dynamic (Basis for Setup, path weight 40)
    Lockout Info:
      Locked Out: No

  Config Parameters:
    Bandwidth: 200      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
```

```
Path-selection Tiebreaker:
  Global: not set   Tunnel Specific: not set   Effective: min-fill (default)
Hop Limit: disabled
Cost Limit: 50
```

Note The accumulated path cost for co-routed bidirectional LSP includes both, forward and reverse path cost.

Flex LSP ECMP min-fill, max-fill, random

MPLS TE chooses the least utilized path given the same accumulated path cost and hop count. However, in Cisco IOS XE Everest 16.5.1, you can steer traffic to an alternate path, depending on:

Tie-breaker 1: min-fill (choose least-utilized path)

- Smaller path cost value
- Larger path minimum bandwidth
- Smaller hop count value
- Lexicographically selects a greater path bandwidth vector (bandwidth value is sorted in ascending order)

Tie-breaker 2: max-fill (choose most-utilized path)

- Smaller path cost value
- Smaller path minimum bandwidth
- Smaller hop count value
- Lexicographically selects a smaller path bandwidth vector (bandwidth value is sorted in ascending order)

Tie-breaker 3: random

- Smaller path cost value
- Smaller hop count value
- Randomly selects a path regardless of the path bandwidth value (as long as it satisfies the request constraints)

The key decision criterion is still path cost; the tie-breakers evaluate, in sequence, until a path is selected. The random tie-breaker does not take the available link bandwidth into consideration.

Configuring Flex LSP ECMP min-fill and max-fill

ECMP can be configured globally or specifically to a tunnel. Tunnel specific configuration override the global tiebreaker selection. The default selection algorithm is min-fill.

For a Flex LSP co-routed tunnel, the same path-selection tiebreaker should be configured on each end of the bi-directional LSP.

Global Configuration Command

```
Router(config)# mpls traffic-eng path-selection tiebreaker ?
  max-fill  Use max-fill tiebreaker
```

```
  min-fill  Use min-fill tiebreaker
  random    Use random tiebreaker
```

Use the **no mpls traffic-eng path-selection tiebreaker** command to remove the global configuration.

Tunnel-Specific Configuration Commands

Configure path selection tiebreaker that is applicable to a specific tunnel. Tunnel specific configuration override global tiebreaker configuration.

```
Router(config)# interface tunnel 1400
Router(config-if)# tunnel mpls traffic-eng path-selection tiebreaker ?
  max-fill  Use max-fill tiebreaker
  min-fill  Use min-fill tiebreaker
  random    Use random tiebreaker
```

Use the **no tunnel mpls traffic-eng path-selection tiebreaker** command to remove the tunnel-specific configuration.

```
Router(config)# interface tunnel 2000
Router(config-if)# ip unnumbered loopback0
Router(config-if)# tunnel mode mpls traffic-eng
Router(config-if)# tunnel destination 56.6.6.6
Router(config-if)# tunnel mpls traffic-eng bidir association type co-routed fault-oam wrap
Router(config-if)# tunnel mpls traffic-eng bidir association id 2000 source 55.5.5.5 global
 2000
Router(config-if)# tunnel mpls traffic-eng path-select tie min-fill
Router(config-if)# tunnel mpls traffic-eng bandwidth 5000
Router(config-if)# tunnel mpls traffic-eng autoroute announce
Router(config-if)# tunnel mpls traffic-eng path-option 10 dynamic
```

```
Router(config)# interface tunnel 2000
Router(config-if)# ip unnumbered loopback0
Router(config-if)# tunnel mode mpls traffic-eng
Router(config-if)# tunnel destination 56.6.6.6
Router(config-if)# tunnel mpls traffic-eng bidir association type co-routed fault-oam wrap
Router(config-if)# tunnel mpls traffic-eng bidir association id 2000 source 55.5.5.5 global
 2000
Router(config-if)# tunnel mpls traffic-eng path-select tie max-fill
Router(config-if)# tunnel mpls traffic-eng bandwidth 5000
Router(config-if)# tunnel mpls traffic-eng autoroute announce
Router(config-if)# tunnel mpls traffic-eng path-option 10 dynamic
```

Verifying the Flex LSP ECMP min-fill and max-fill

Global Commands

To display the current global tiebreaker value.

- If the global tiebreaker is not set, the effective tiebreaker is the default setting (min-fill).
- If the global tiebreaker is set, it is the effective tiebreaker as well.

```
Router# show mpls traffic-eng path-selection tiebreaker
```

```
Global Path Selection Tiebreaker
---------------------------------------------
Configured Tiebreaker: not set
Effective Tiebreaker: min-fill (default)
```

Tunnel-Specific Commands

To display current tiebreaker configuration of a tunnel:

```
Router# show mpls traffic-eng tunnels tunnel 1400 path-selection tiebreaker
Tunnel Name                  Global     Tunnel Specific      Effective
Router_t1400                 not set    max-fill             max-fill
```

To display the tiebreaker:

```
Router# show mpls traffic-eng tunnels tunnel 2000 detail

Name: Router_t2000                         (Tunnel2000) Destination: 56.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, type dynamic (Basis for Setup, path weight 80)
    Lockout Info:
      Locked Out: No

  Config Parameters:
    Bandwidth: 5000     kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    Path-selection Tiebreaker:
      Global: not set   Tunnel Specific: min-fill   Effective: min-fill
    Hop Limit: 5
    Cost Limit: disabled
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 5000 [400000] bw-based
    auto-bw: disabled
    Association Type: Single Sided Bidirectional LSPs, Co-routed: YES
```

Use the **debug mpls traffic-eng path ecmp** command to display the ECMP-related debug information, such as for BWV comparison when all other tiebreakers failed to yield a selected path. Use this command in conjunction with the **debug mpls traffic-eng path spf** or **debug mpls traffic-eng path lookup** commands.

Restore Path Option

The restore path option signals a restore LSP after the double failure of both, primary and protect LSPs. The restore LSP is signaled only after both, primary and protect LSPs fail or are administratively down and it is destroyed when the primary LSP comes back up. If the sticky option is configured, and both, primary and protect LSPs fail, restore LSP is destroyed when either the primary or protect LSP comes up. Also, restore LSP can be SRLG capable if it is configured.

Note Traffic loss is expected and acceptable until the restore LSP is signaled and established.

When the primary LSP recovers, traffic is automatically switched back to the primary LSP (make-before-break scenario) and restore LSP is torn down. If the sticky option is configured in the path option, protect LSP can recover before the primary LSP. In this case, traffic reverted to the protect LSP and the restore LSP is torn down.

To enable restore LSP for a tunnel, configure the restore path option under the tunnel interface using the command:

```
tunnel mpls traffic-eng path-option restore <index> [dynamic | explicit]
```

Multiple restore path options can be configured on a tunnel. However, only the restore path option with the same index as the path option used by primary LSP is used for establishing the restore LSP. For example, after primary and protect LSPs associated with the path option index X are administratively down, restore LSP is signaled using the restore path option with index X, if it is configured. Else, no restore LSP is signaled.

Note Restore path option is supported for unidirectional, bidirectional co-routed, or non co-routed tunnels. It is not supported for lockout, sticky, and non-revertive modes.

Configuring the Restore Path Option

```
tunnel mpls traffic-eng path-option restore <index> [dynamic | explicit]
Router(config-if)# tunnel mpls traffic-eng path-option ?
  <1-1000> preference for this path option
  protect   a path protection setup option
  restore   a path restore setup option
```

Verifying the Restore Path Option

The following is an example of the information displayed for the tunnel when restore LSP is carrying traffic:

```
Router# show mpls traffic-eng tunnels Tunnel 100
Name: iolA_t100                           (Tunnel100) Destination: 192.168.1.4
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path restore option 1, type dynamic (Basis for Restore, path weight 40)
    path option 1, (LOCKDOWN & STICKY) type dynamic
    Path Protection: Requested
    path protect option 1, (LOCKDOWN & STICKY) type diverse
    Path Restore: Restore lsp in use.
    path restore option 1, type dynamic (Basis for Restore, path weight 40)
    Lockout Info:
      Locked Out: No
```

```
Router# show running-config interface tunnel 1000
Building configuration...

Current configuration: 675 bytes
!
interface Tunnel1000
ip unnumbered Loopback0
tunnel mode mpls traffic-eng
tunnel destination 56.6.6.6
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 200
tunnel mpls traffic-eng path-option 10 dynamic lockdown sticky
tunnel mpls traffic-eng path-option protect 10 diverse non-revertive lockdown
tunnel mpls traffic-eng path-option restore 10 dynamic
tunnel mpls traffic-eng bidirectional association id 1000 source-address 55.5.5.5 global-id
 1000
tunnel mpls traffic-eng bidirectional association type co-routed fault-oam wrap-protection
tunnel mpls traffic-eng bfd encap-mode gal flex_bfd
```

```
Router# show mpls traffic-eng tunnels tunnel 1000

Name: Router_t1000 (Tunnel1000) Destination: 56.6.6.6
Status:
Admin: up Oper: up Path: valid Signalling: connected
path restore option 10, type dynamic (Basis for Restore, path weight 6)
path option 10, (LOCKDOWN & STICKY) type dynamic
Path Protection: Requested
path protect option 10, (LOCKDOWN & STICKY) type (NON-REVERTIVE)diverse
Path Restore: Restore lsp in use.
path restore option 10, type dynamic (Basis for Restore, path weight 6)
Lockout Info:
```

```
Locked Out: No

Config Parameters:
Bandwidth: 200 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
Metric Type: TE (default)
Path-selection Tiebreaker:
Global: not set Tunnel Specific: not set Effective: min-fill (default)
Hop Limit: disabled
Cost Limit: disabled
Path-invalidation timeout: 10000 msec (default), Action: Tear
AutoRoute: enabled LockDown: disabled Loadshare: 200 [10000000] bw-based
auto-bw: disabled
Association Type: Single Sided Bidirectional LSPs, Co-routed: YES
Association ID: 1000, Source: 55.5.5.5, Global ID: 1000
Reverse Bandwidth: 200 kbps, Standby: 200 kbps
Fault-OAM: enabled, Path-Protection: no protection, Wrap-Protection: enabled, Wrap-Capable:
 No
BFD:
Mode: GAL
Template: flex_bfd
BFD Counters:
Session create: 14
Session delete: 11
Session create error: 0
Session delete error: 0
Active Path Option Parameters:
State: dynamic path option 10 is active
BandwidthOverride: disabled LockDown: disabled Verbatim: disabled
Node Hop Count: 3

InLabel: -
OutLabel: GigabitEthernet0/3/0, 21
Next Hop: 101.105.1.1
Reverse Associated LSP Information:
Signaled Name: 56.6.6.6 1000
Tunnel: 1000, Source: 56.6.6.6, Dest: 55.5.5.5, LSP: 133 State: Up
Lockout Info:
Locked out: No
Lockout Originated By: None
Association:
Association Type: Single Sided Bidirectional LSPs
Association ID: 1000 Source: 55.5.5.5
Restoring
Extended Association:
Global source: 1000
Extended ID:
0x38060606 (56.6.6.6)
0x00000085 (0.0.0.133)
0x00000000 (0.0.0.0)
0x00000005 (0.0.0.5)
RSVP Signalling Info:
Src 55.5.5.5, Dst 56.6.6.6, Tun_Id 1000, Tun_Instance 91
RSVP Path Info:
My Address: 101.105.1.2
Explicit Route: 101.105.1.1 101.102.6.1 101.102.6.2 102.106.2.1
102.106.2.2 56.6.6.6
Record Route: NONE
Tspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
RSVP Resv Info:
Record Route: NONE
Fspec: ave rate=200 kbits, burst=1000 bytes, peak rate=200 kbits
Shortest Unconstrained Path Info:
Path Weight: 6 (TE)
Explicit Route: 101.105.5.2 101.105.5.1 101.102.6.1 101.102.6.2
```

```
102.106.2.1 102.106.2.2 56.6.6.6
Sticky Primary Path Info:
Path Option Index: 10
Explicit Route: 102.105.12.2 102.105.12.1 102.106.2.1 102.106.2.2
56.6.6.6
Reason for the tunnel being down: Bidirectional: standby error from [56.6.6.6][TAIL] LSP[73]

History:
Tunnel:
Time since created: 34 minutes, 3 seconds
Time since path change: 4 seconds
Number of LSP IDs (Tun_Instances) used: 91
Current LSP: [ID: 91]
Uptime: 9 seconds
Selection:
Prior LSP: [ID: 77]
ID: path option 10 [102]
Removal Trigger: path verification failed
Last Error: CTRL:: Explicit path has unknown address, 102.105.12.2
```