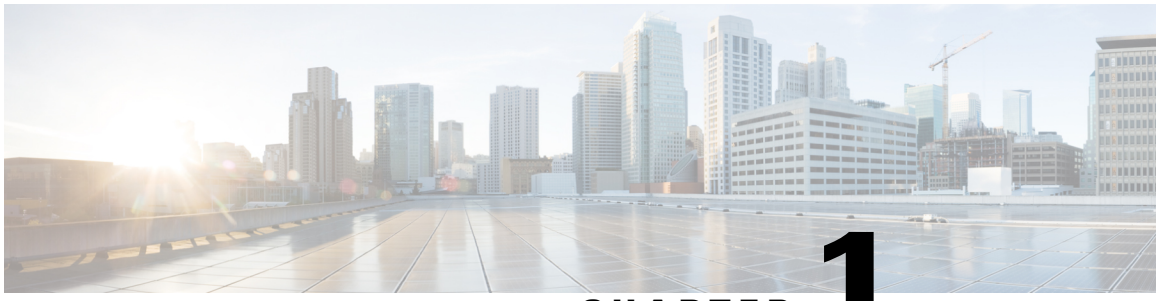
MPLS Basic MPLS Configuration Guide, Cisco IOS XE Fuji 16.8.x

First Published: 2018-03-30

Last Modified: 2018-03-30

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

Read Me First

Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E (for Catalyst Switching) and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

Feature Information

Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

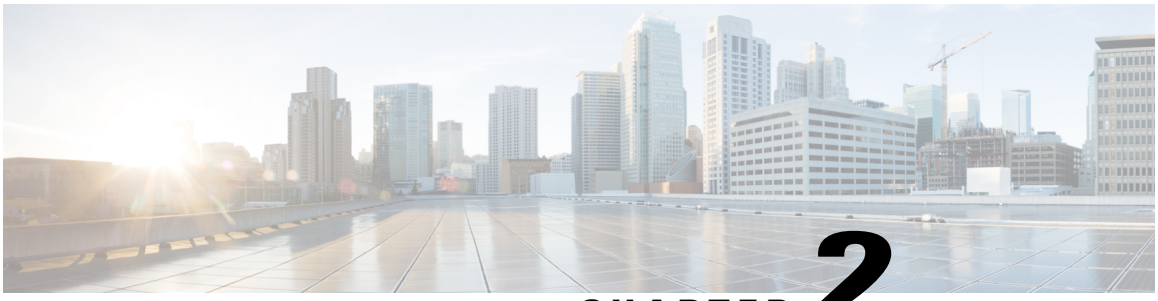
Related References

- Cisco IOS Command References, All Releases

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the What's New in Cisco Product Documentation RSS feed. RSS feeds are a free service.



CHAPTER 2

MPLS Transport Profile

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.

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.

Restrictions for MPLS Transport Profile

- 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.
- Ethernet subinterfaces are not supported.

- IPv6 addressing is not supported.

L2VPN Restrictions

- 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.
- Static pseudowire Operations, Administration, and Maintenance (OAM) protocol and BFD VCCV attachment circuit (AC) status signaling are mutually exclusive protocols. Bidirectional Forwarding Detection (BFD) and Virtual Circuit Connectivity Verification (VCCV) in failure detection mode can be used with Static Pseudowire OAM protocol.
- BFD VCCV AC status signaling cannot be used in pseudowire redundancy configurations. You can use Static Pseudowire OAM instead.

Ping and Trace Restrictions

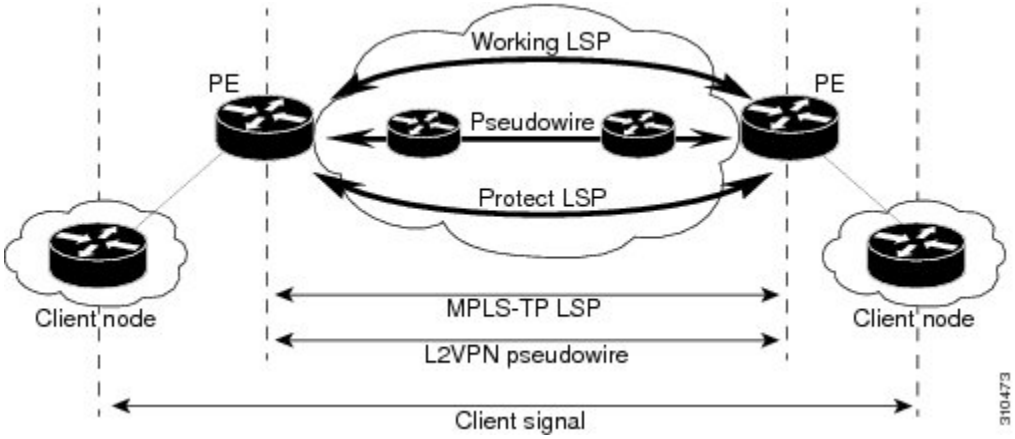
- Ping for static pseudowires over MPLS-TP tunnels is not supported.
- Pseudowire ping and traceroute functionality for multisegment pseudowires that have one or more static pseudowire segments is not supported.
- The following packet format is supported:
    - A labeled packet with Generic Associated Channel Label (GAL) at the bottom of the label stack.
    - ACH channel is IP (0x21).
    - RFC-4379-based IP, UDP packet payload with valid source.
    - Destination IP address and UDP port 3503.
- Default reply mode for (1) is 4—Reply via application level control channel is supported. An echo reply consists of the following elements:
    - A labeled packet with a GAL label at the bottom of the label stack.
    - Associated Channel (ACh) is IP (0x21).
    - RFC-4379-based IP, UDP packet payload with valid source.
    - Destination IP address and UDP port 3503.
- The optional “do not reply” mode may be set.
- The following reply modes are not allowed and are disabled in CLI:
    - 2—Reply via an IPv4/IPv6 UDP packet
    - 3—Reply via an IPv4/IPv6 UDP packet with router alert

- Force-explicit-null is not supported with ping and trace.
- Optional Reverse Path Connectivity verification is not supported.

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 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.

Support for MPLS Transport Profile OAM

Several Operations, Administration, and Maintenance (OAM) protocols and messages support the provisioning and maintenance of Multiprotocol Label Switching Transport Profile (MPLS-TP) tunnels and bidirectional label switched paths (LSPs).

The following OAM messages are forwarded along the specified MPLS LSP:

- OAM Fault Management—Alarm Indication Signal (AIS), Link Down Indication (LDI), and Lock Report (LKR) messages (GAL with BFD messages).
- OAM Connection Verification—Ping and traceroute messages (GAL with IP channel by default).
- OAM Continuity Check—Bidirectional Forwarding Detection (BFD) messages—non-IP BFD and IP BFD (GAL with non-IP BFD channel or IP BFD channel depending on message format).
- The following messages are forwarded along the specified pseudowire:
  - Static pseudowire OAM messages
  - Pseudowire ping and traceroute messages
  - BFD messages
- MPLS-TP OAM Fault Management (LDI, AIS, and LKR messages)—LDI messages are AIS messages whose L-flags are set. The LDI messages are generated at midpoint nodes when a failure is detected. From the midpoint, an LDI message is sent to the endpoint that is reachable with the existing failure. Similarly, LKR messages are sent from a midpoint node to the reachable endpoint when an interface is administratively shut down. By default, the reception of LDI and LKR messages on the active LSP at an endpoint will cause a path protection switchover, whereas the reception of an AIS message will not.
- MPLS-TP OAM Fault Management with Emulated Protection Switching for LSP Lockout—Cisco implements a form of Emulated Protection Switching to support LSP Lockout using customized Fault messages. When a Lockout message is sent, it does not cause the LSP to be administratively down. The Cisco Lockout message causes a path protection switchover and prevents data traffic from using the LSP. The LSP remains administratively up so that BFD and other OAM messages can continue to traverse it and so that maintenance of the LSP can take place (such as reconfiguring or replacing a midpoint LSR). After OAM verifies the LSP connectivity, the Lockout is removed and the LSP is brought back to service. Lockout of the working LSP is not allowed if a protect LSP is not configured. Conversely, the Lockout of a protect LSP is allowed if a working LSP is not configured.
- LSP ping and trace—To verify MPLS-TP connectivity, use the **ping mpls tp** and **trace mpls tp** commands. You can specify that echo requests be sent along the working LSP, the protect LSP, or the active LSP. You can also specify that echo requests be sent on a locked-out MPLS-TP tunnel LSP (either working or protected) if the working or protected LSP is explicitly specified. You can also specify ping/trace messages with or without IP.
- MPLS-TP OAM Continuity Check (CC) via BFD and Remote Defect Indication (RDI)—RDI is communicated via the BFD diagnostic field in BFD CC messages. BFD sessions run on both the working LSP and the protect LSP. To perform a path protection switchover within 60 milliseconds on an MPLS-TP endpoint, use the BFD Hardware Offload feature, which enables the router hardware to construct and send BFD messages, removing the task from the software path. The BFD Hardware Offload feature is enabled automatically on supported platforms.

MPLS-TP OAM GACH—Generic Associated Channel (G-ACh) is the control channel mechanism associated with Multiprotocol Label Switching (MPLS) LSPs in addition to MPLS pseudowire. The G-ACh Label (GAL) (Label 13) is a generic alert label to identify the presence of the G-ACh in the label packet. It is taken from the reserved MPLS label space. G-ACh/GAL supports OAMs of LSPs and in-band OAMs of pseudowires (PWs). OAM messages are used for fault management, connection verification, continuity check, and so on.

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. You can acquire per pseudowire OAM for attachment circuit/pseudowire notification over the VCCV channel with or without the control word.

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 **mpls tp link** command is used to associate 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 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.

See the section Configuring MPLS-TP Links and Physical Interfaces, on page 29, for more information.

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

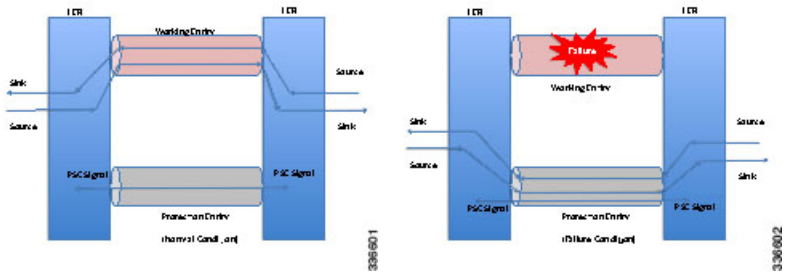
The Multiprotocol Label Switching (MPLS) Transport Profile (TP) enables you to create tunnels that provide the transport network service layer over which IP and MPLS traffic traverse.

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.

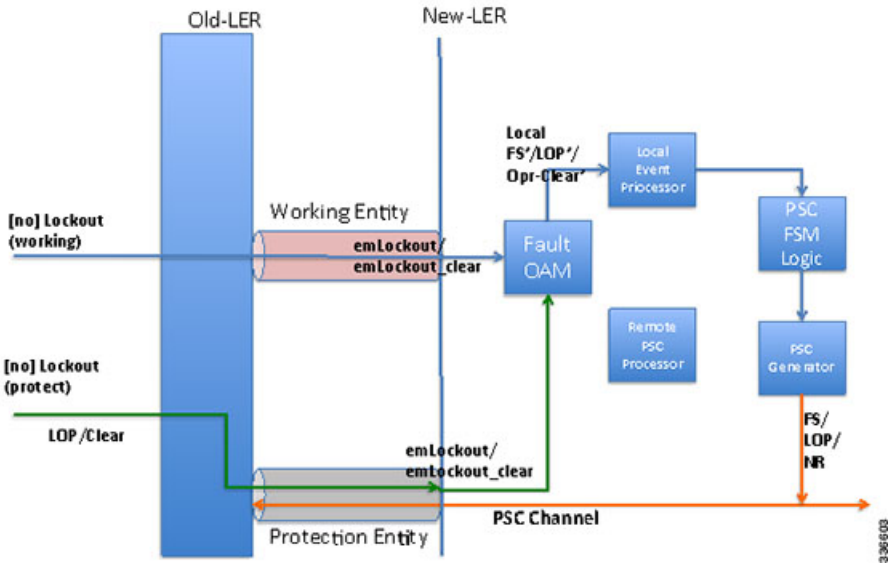
An endpoint implementing the newer PSC version may have to communicate with another endpoint implementing an older version. Because there is no mechanism to exchange the capabilities, the PSC implementation must interoperate with another peer endpoint implementing emulated APS. In this scenario, the 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:

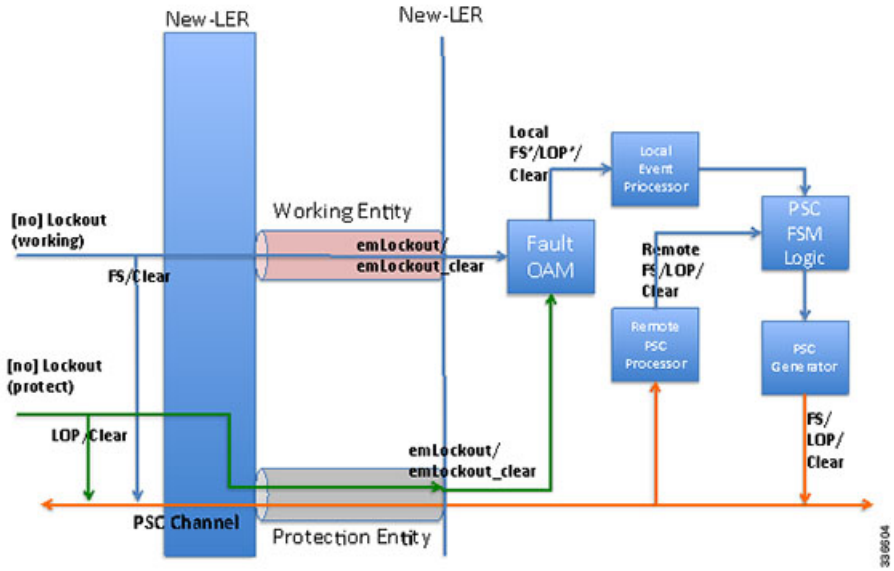
- New-old implementation.
- New-new implementation.

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:

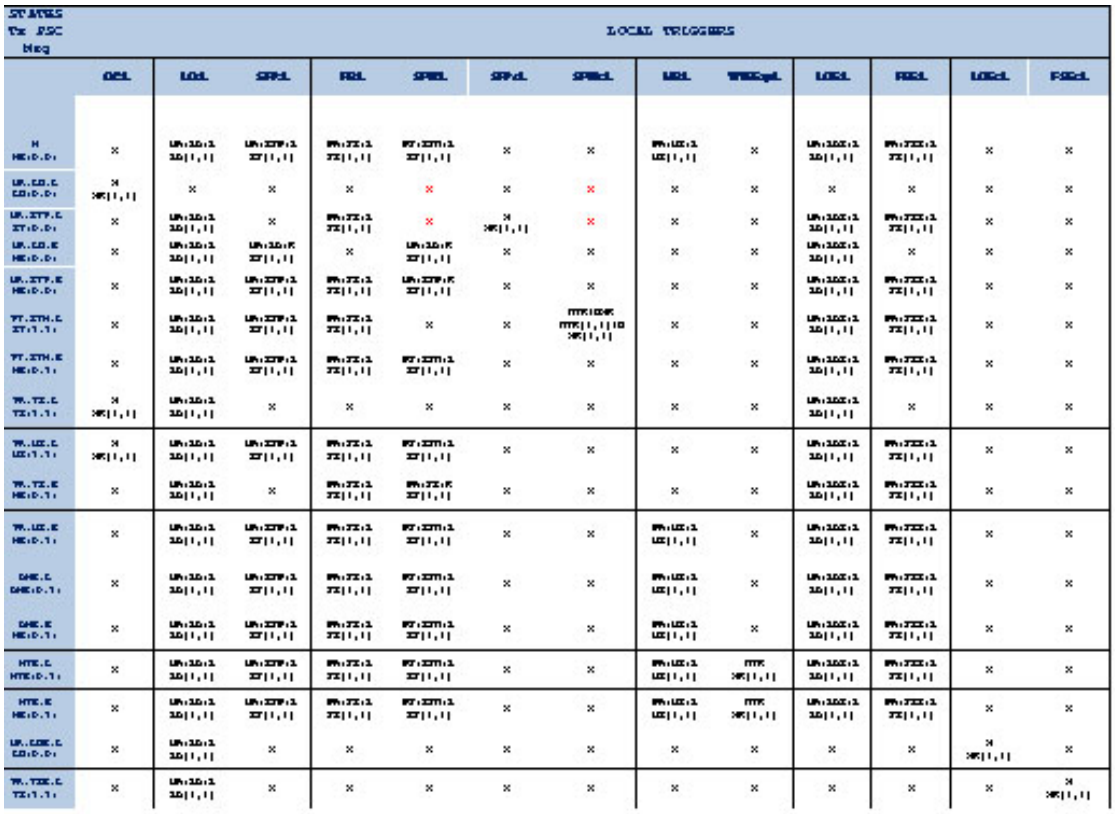
1. Opr-Clear (Operator Clear)
2. LoP (Lockout of Protection)
3. LoP'/LoP'-Clear
4. FS (Forced Switch)
5. FS'/FS'-Clear
6. MS (Manual-Switch)

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.

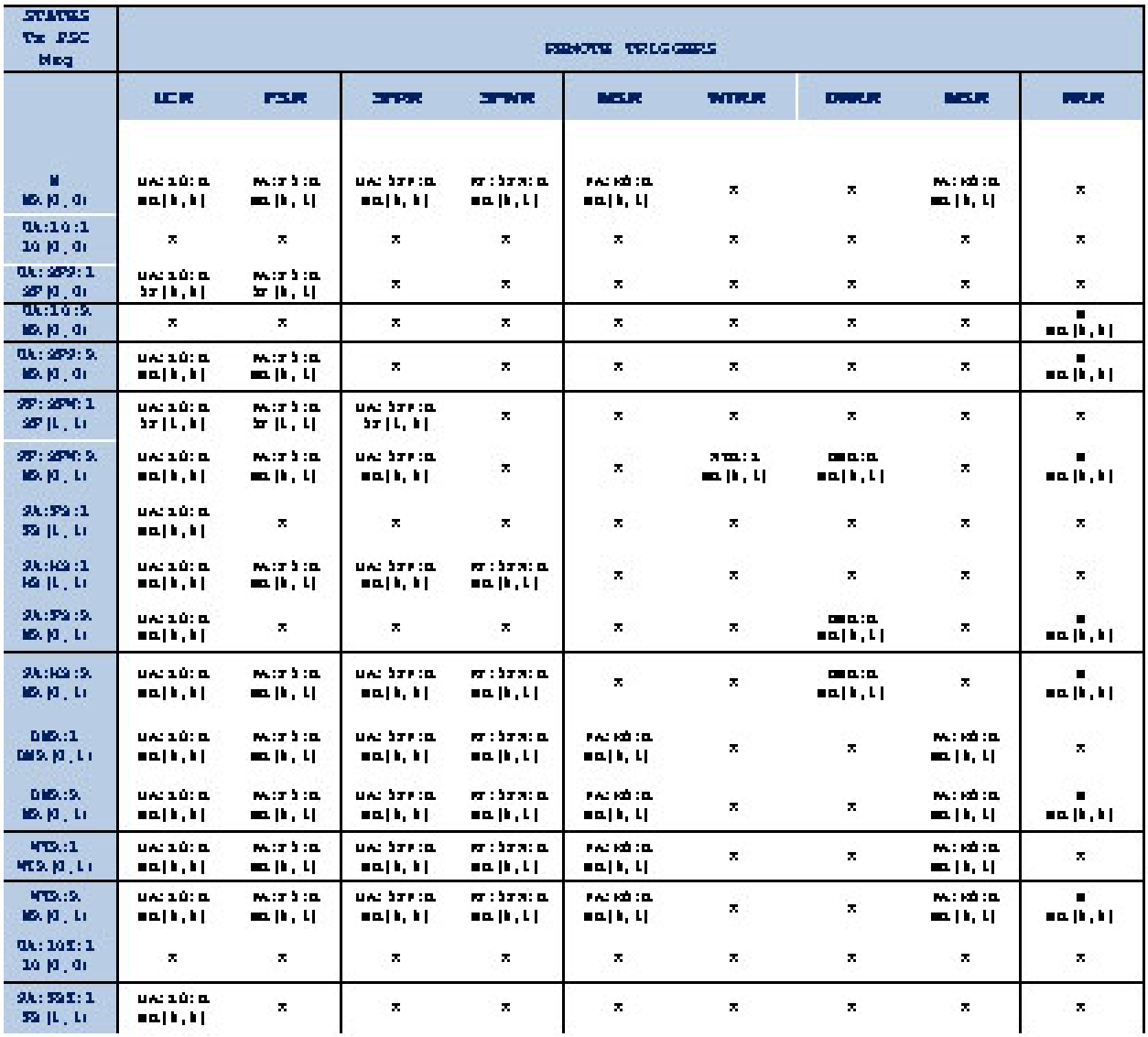
The priority definition for Signal Fail (SF), Signal Degrade (SD), Manual Switch (MS), WTR, Do Not Revert (DNR), and No Request (NR) remains unchanged.

PSC Finite State Machine Logic

The PSC implementation follows the state transition logic defined in the following tables:



336509



The PSC finite state machine (FSM) consists of the following states used in the above tables:

1. Normal state.
2. UA:LO:L Protect is unavailable because of a lockout protection issued locally.
3. UA:LOE:L Protect is unavailable because of receipt of emLockout on the protected LSP.
4. UA:LO:R Protect is unavailable because of a lockout of protection issued remotely.
5. UA:SFP:L Protect is unavailable because of a local sgnal fail on the protected LSP.
6. UA:SFP:R Protect is unavailable because of a remote signal fail on the protected LSP.
7. PF:SFW:L Protecting failure because of a local signal fail on the working LSP.
8. PF:SFW:R Protecting failure because of a remote signal fail on the working LSP.
9. PA:FS:L Protecting administrative because of a local force switch (FS).
10. PA:FS:R Protecting administrative because of a remote FS.

11. PA:FSE:R Protecting administrative because of a receipt of emLockout on the working LSP.
12. PA:MS:L Protecting administrative because of a local manual switch.
13. PA:MS:R Protecting administrative because of a remote manual switch.
14. WTR:L Local wait-to-restore (WTR) state.
15. WTR:R Remote WTR state.
16. DNR:L Local do-not-revert (DNR) state.
17. DNR:R Remote DNR state.

The following are the PSC FSM events based on priority (higher to lower):

1. OC:L Local operator command cleared.
2. LO:L Local lockout of protect command.
3. LOEc:L Receipt of emLockout clear of protect.
4. LOE:L Receipt of emLockout on the protected LSP.
5. LO:R Remote lockout of protection.
6. FS:L Local FS.
7. FSEc:L Receipt of emLockout clear of the working LSP.
8. FSE:L Receipt of emLockout of the working LSP.
9. FS:R Remote FS.
10. SFP:L Local signal fail on the protected LSP.
11. SFP:R Remote signal fail on the protected LSP.
12. SFW:L Local signal fail on the working LSP.
13. SFW:R Remote signal fail on the working LSP.
14. SFPc:L Local signal fail on protect cleared.
15. SFWc:L Local signal fail on the working cleared.
16. MS:L Local manual switch.
17. MS:R Remote manual switch.
18. WTRExp:L Local WTR timer expired.
19. WTR:R Remote WTR event.
20. DNR:R Remote DNR event.
21. NR:R Remote NR event.

The signal-degrade event on the working/protected LSP is not supported.

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.

	Command or Action	Purpose
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** **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 MPLS-TP configuration mode, from which you can configure MPLS-TP parameters for the device.

	Command or Action	Purpose
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, OAM class, and VCCV BFD template.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *class-name*
4. **encapsulation mpls**
5. **control-word**
6. **protocol** {**l2tpv2** | **l2tpv3** | **none**} [*l2tp-class-name*]
7. **preferred-path** {**interface tunnel** *tunnel-number* | **peer** {*ip-address* | *host-name*}} [**disable-fallback**]
8. **status protocol notification static** *class-name*
9. **vccv bfd template** *name* [**udp** | **raw-bfd**]
10. **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	**protocol** {**l2tpv2** | **l2tpv3** | **none**} [*l2tp-class-name*] **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	**vccv bfd template** *name* [**udp** | **raw-bfd**] **Example:** `Device(config-pw-class)# vccv bfd template bfd-temp1 raw-bfd`	Specifies the VCCV BFD template to use.
Step 10	**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** *type number*
4. **xconnect** *peer-ip-address vc-id* {**encapsulation** {**l2tpv3** [**manual**] | **mpls** [**manual**]} | **pw-class** *pw-class-name*} [**pw-class** *pw-class-name*] [**sequencing** {**transmit** | **receive** | **both**}]
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** *type number* **Example:** `Device(config)# interface Ethernet 1/0`	Specifies the interface and enters interface configuration mode.
Step 4	**xconnect** *peer-ip-address vc-id* {**encapsulation** {**l2tpv3** [**manual**] | **mpls** [**manual**]} | **pw-class** *pw-class-name*} [**pw-class** *pw-class-name*] [**sequencing** {**transmit** | **receive** | **both**}] **Example:** `Device(config-if)# xconnect 10.131.191.251 100 encapsulation mpls manual pw-class mpls-tp-class1`	Binds the attachment circuit to a pseudowire VC and enters xconnect interface configuration mode.

	Command or Action	Purpose
Step 5	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-if-xconn)# mpls label 100 150`	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 bandwidth** *num*
7. **tp source** *node-id* [*global-id num*]
8. **tp destination** *node-id* [**tunnel-tp** *num*[**global-id** *num*]]
9. **bfd** *bfd-template*
10. **working-lsp**
11. **in-label** *num*
12. **out-label** *num* **out-link** *num*
13. **exit**
14. **protect-lsp**
15. **in-label** *num*
16. **out-label** *num* **out-link** *num*
17. **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`	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.

	Command or Action	Purpose
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 bandwidth** *num* **Example:** `Device(config-if)# tp bandwidth 10000`	Specifies the tunnel bandwidth.
Step 7	**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 8	**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 9	**bfd** *bfd-template* **Example:** `Device(config-if)# bfd mpls-tp-bfd-2`	Specifies the BFD template.
Step 10	**working-lsp** **Example:** `Device(config-if)# working-lsp`	Specifies a working LSP, also known as the primary LSP.
Step 11	**in-label** *num* **Example:** `Device(config-if-working)# in-label 111`	Specifies the in-label number.
Step 12	**out-label** *num* **out-link** *num* **Example:** `Device(config-if-working)# out-label 112 out-link`	Specifies the out-label number and out-link.

	Command or Action	Purpose
Step 13	**exit** **Example:** `Device(config-if-working)# exit`	Exits working LSP interface configuration mode and returns to interface configuration mode.
Step 14	**protect-lsp** **Example:** `Device(config-if)# protect-lsp`	Specifies a backup for a working LSP.
Step 15	**in-label** *num* **Example:** `Device(config-if-protect)# in-label 100`	Specifies the in label.
Step 16	**out-label** *num* **out-link** *num* **Example:** `Device(config-if-protect)# out-label 113 out-link`	Specifies the out label and out link.
Step 17	**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. **bandwidth** *num*
6. **in-label** *num* **out-label** *num* **out-link** *num*
7. **exit**
8. **reverse-lsp**
9. **bandwidth** *num*
10. **in-label** *num* **out-label** *num* **out-link** *num*
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	**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 2 tunnel-tp 4 lsp protect destination 10.11.11.11 global-id 11 tunnel-tp 12`	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	**bandwidth** *num* **Example:** `Device(config-mpls-tp-lsp-forw)# bandwidth 100`	Specifies the bandwidth.
Step 6	**in-label** *num* **out-label** *num* **out-link** *num* **Example:** `Device(config-mpls-tp-lsp-forw)# in-label 53 out-label 43 out-link 41`	Specifies the in label, out label, and out link numbers.
Step 7	**exit** **Example:** `Device(config-mpls-tp-lsp-forw)# exit`	Exits MPLS-TP LSP forward LSP configuration mode.
Step 8	**reverse-lsp** **Example:** `Device(config-mpls-tp-lsp)# reverse-lsp`	Enters MPLS-TP LSP reverse LSP configuration mode.
Step 9	**bandwidth** *num* **Example:** `Device(config-mpls-tp-lsp-rev)# bandwidth 100`	Specifies the bandwidth.
Step 10	**in-label** *num* **out-label** *num* **out-link** *num* **Example:** `Device(config-mpls-tp-lsp-rev)# in-label 33 out-label 23 out-link 44`	Specifies the in-label, out-label, and out-link numbers.
Step 11	**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*} **rx-mac** *mac-address*
6. **ip rsvp bandwidth** [**rdm** [**bc0** *interface-bandwidth*] [[*single-flow-bandwidth* [**bc1** *bandwidth* | **sub-pool** *bandwidth*]]] [*interface-bandwidth* [*single-flow-bandwidth* [**bc1** *bandwidth* | **sub-pool** *bandwidth*]] | **mam max-reservable-bw** [*interface-bandwidth* [*single-flow-bandwidth*] [**bc0** *interface-bandwidth* [**bc1** *bandwidth*]]] | **percent** *percent-bandwidth* [*single-flow-bandwidth*]]
7. **end**
8. **show mpls tp link-numbers**

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 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*} **rx-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

	Command or Action	Purpose
		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	**ip rsvp bandwidth** [**rdm** [**bc0** *interface-bandwidth*] [[*single-flow-bandwidth* [**bc1** *bandwidth* | **sub-pool** *bandwidth*]]] [*interface-bandwidth* [*single-flow-bandwidth* [**bc1** *bandwidth* | **sub-pool** *bandwidth*]] | **mam max-reservable-bw** [*interface-bandwidth* [*single-flow-bandwidth*] [**bc0** *interface-bandwidth* [**bc1** *bandwidth*]]] | **percent** *percent-bandwidth* [*single-flow-bandwidth*]] **Example:** `Device(config-if)# ip rsvp bandwidth 1158 100`	Enables Resource Reservation Protocol (RSVP) bandwidth for IP on an interface. For the Cisco 7600 platform, if you configure non-zero bandwidth for the TP tunnel or at a midpoint LSP, make sure that the interface to which the output link is attached has enough bandwidth available. For example, if three tunnel LSPs run over link 1 and each LSP was assigned 1000 with the **tp bandwidth** command, the interface associated with link 1 needs bandwidth of 3000 with the **ip rsvp bandwidth** command.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 8	**show mpls tp link-numbers** **Example:** `Device# show mpls tp link-numbers`	Displays the configured links.

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *name* **point-to-point**
4. **neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*}
5. **mpls label** *local-pseudowire-label remote-pseudowire-label*
6. **mpls control-word**
7. **neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*}
8. **mpls label** *local-pseudowire-label remote-pseudowire-label*
9. **mpls control-word**
10. **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	**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 5	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-vfi)# mpls label 101 201`	Configures the static pseudowire connection by defining local and remote circuit labels.
Step 6	**mpls control-word** **Example:** `Device(config-vfi)# mpls control-word`	Specifies the control word.
Step 7	**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 8	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-vfi)# mpls label 102 202`	Configures the static pseudowire connection by defining local and remote circuit labels.
Step 9	**mpls control-word** **Example:** **Example:** `Device(config-vfi)# mpls control-word`	Specifies the control word.
Step 10	**end** **Example:** `Device(config)# end`	Exits VFI configuration mode and returns to privileged EXEC mode.

Configuring a Template with Pseudowire Type-Length-Value Parameters

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-tlv template** *template-name*
4. **tlv** [*type-name*] *type-value length* [**dec** | **hexstr** | **str**] *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	**pseudowire-tlv template** *template-name* **Example:** `Device(config)# pseudowire-tlv template statictemp`	Creates a template of pseudowire type-length-value (TLV) parameters and enters pseudowire TLV template configuration mode.
Step 4	**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 5	**end** **Example:** `Device(config-pw-tlv-template)# end`	Exits pseudowire TLV template 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.

	Command or Action	Purpose
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).
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.

	Command or Action	Purpose
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-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. **protocol** {**l2tpv2** | **l2tpv3** | **none**} [*l2tp-class-name*]
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.

	Command or Action	Purpose
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	**protocol** {**l2tpv2** | **l2tpv3** | **none**} [*l2tp-class-name*] **Example:** `Device(config-pw-class)# protocol none`	Specifies the type of protocol. Use the **protocol none** command to specify a static pseudowire.
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*}	Sets up an emulated VC and enters VFI neighbor configuration mode.

	Command or Action	Purpose
	Example: `Device(config-vfi)# neighbor 10.111.111.111 123 pw-class atom`	**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:** `Device(config-vfi-neighbor)# mpls label 101 201`	Configures the static pseudowire connection by defining local and remote circuit labels.
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.

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

The following example shows how to configure static-to-dynamic multisegment pseudowires for Layer 2 VFI.

```
l2 vfi atom point-to-point (static-dynamic MSPW)
 neighbor 10.116.116.116 4294967295 pw-class dypw (dynamic)
  neighbor 10.111.111.111 123 pw-class stpw (static)
  mpls label 101 201
  mpls control-word
  local interface 4
  tlv mtu 1 4 1500
  tlv description 3 6 str abcd
  tlv descr C 4 hexstr 0505
```

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
```

Additional References for MPLS Transport Profile

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

Standards and RFCs

Standard/RFC	Title
draft-ietf-mpls-tp-gach-gal-xx	*MPLS Generic Associated Channel*
RFC 5586	*MPLS Generic Associated Channel*
RFC 5885	*Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)*
RFC 5921	*A Framework for MPLS in Transport Networks*

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 MPLS Transport Profile

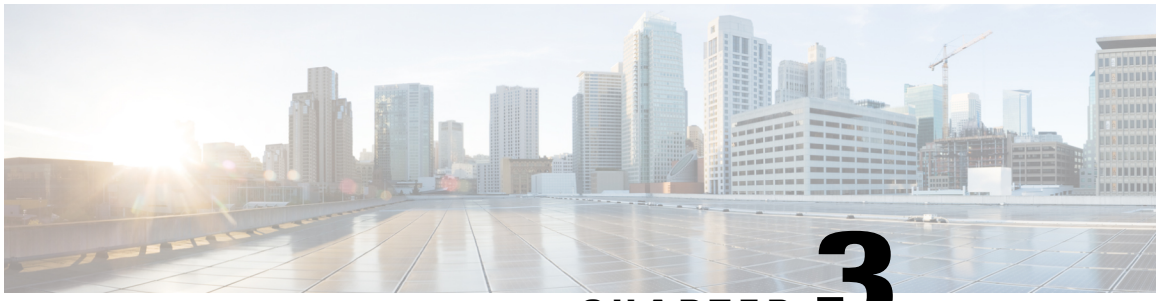
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train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

Feature Name	Releases	Feature Information
MPLS Transport Profile • Bidirectional MPLS-TP LSP • L2VPN Static to Dynamic PW Interconnection & PW Preferred Path for MPLS-TP Tunnels • MPLS TP: IP-less Configuration of MPLS TP Tunnels • MPLS-TP OAM: Continuity Check via BFD • MPLS-TP OAM: Fault Management • MPLS-TP OAM: GACH • MPLS-TP Path Protection • MPLS-TP OAM: Ping/Trace • MPLS-TP: PW Redundancy for Static PWs	Cisco IOS XE Release 3.5S	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 SONET and SDH TDM technologies to packet switching to support services with high bandwidth requirements, such as video. In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router. The following commands were introduced or modified: **debug mpls l2transport static-oam**, **debug mpls tp**, **interface tunnel-tp interval local**, **interface logging (MPLS-TP)**, **medium p2p**, **mpls tp**, **mpls tp link**, **mpls tp lsp ping**, **notification static timeout refresh**, **pseudowire-static-oam class**, **pseudowire-tlv template**, **show mpls l2transport static-oam**, **show mpls tp status protocol**, **tlv**, **tlv template trace mpls tp**.
MPLS Transport Profile • MPLS-TP L2VPN Support for MPLS Transport Profile • MPLS-TP OAM: Continuity Check via BFD • MPLS-TP OAM: Fault Management • MPLS-TP OAM: GACH • MPLS-TP Path Protection • MPLS-TP OAM: Ping/Trace	Cisco IOS XE Release 3.10S	In Cisco IOS XE Release 3.10S, support was added for the Cisco ASR 1000 Router.

Feature Name	Releases	Feature Information
MPLS-TP Linear Protection with PSC Support	Cisco IOS XE Release 3.9S	In Cisco IOS XE Release 3.9S, support was added for the Cisco ASR 903 Router. The following commands were introduced or modified: **[no] psc {fast | slow | remote} refresh interval {***time-in-msec* |*time-in-sec***} [message-count***num***]**, **emulated-lockout**, **manual-switch**, **show mpls tp summary**, **show mpls tp link-numbers**, **debug mpls tp psc packet**, **debug mpls tp psc event**, **clear mplsl tp [tunnel-tp** *tun-num*| **tunnel-name** *name***] psc counter**, **clear mpls tp [tunnel-tp** *tun-num*| **tunnel-name** *name***] psc remote-event**.



CHAPTER 3

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 as to which label value to use 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. This awareness of label bindings by neighboring routers is facilitated by 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)--Used to support MPLS forwarding along normally routed paths
- Resource Reservation Protocol (RSVP)--Used to support MPLS traffic engineering
- Border Gateway Protocol (BGP)--Used to support MPLS virtual private networks (VPNs)

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.

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.

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:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
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 2: 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.

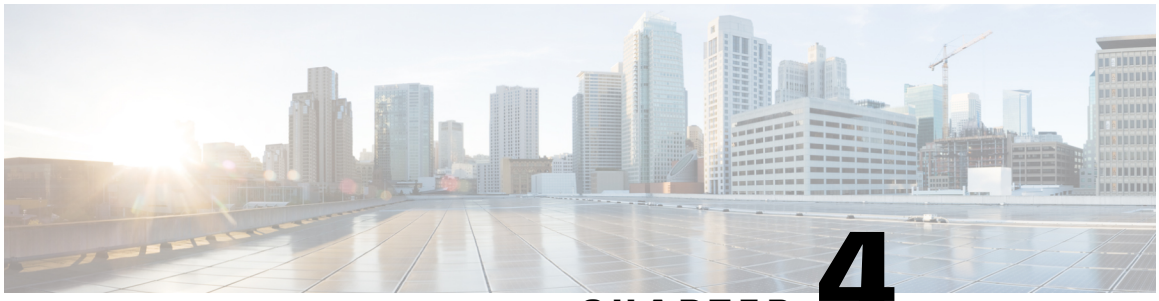
Routing Information Base --See RIB.

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 4

MPLS Infrastructure Changes Introduction of MFI and Removal of MPLS LSC and LC-ATM Features

This document explains the new MPLS Forwarding Infrastructure (MFI) and removal of support for MPLS label switch controller (LSC) and label-controlled ATM (LC-ATM) features and commands.

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 Infrastructure Changes

Introduction of the MPLS Forwarding Infrastructure

The MPLS control plane software is enhanced to make MPLS more scalable and flexible. The MFI, which manages MPLS data structures used for forwarding, replaces the Label Forwarding Information Base (LFIB).

Note The MFI and LFIB do not coexist in the same image. For a list of supported releases, see the "Feature Information for MPLS Forwarding Infrastructure."

Introduction of IP Rewrite Manager

Cisco software introduces a module called the MPLS IP Rewrite Manager (IPRM) that manages the interactions between Cisco Express Forwarding, the IP Label Distribution Modules (LDMs), and the MFI. MPLS IPRM is enabled by default. You need not configure or customize the IPRM. These commands are related to IPRM:

- **clear mpls ip iprm counters**
- **debug mpls ip iprm**
- **debug mpls ip iprm cef**
- **debug mpls ip iprm events**
- **debug mpls ip iprm ldm**
- **debug mpls ip iprm mfi**
- **show mpls ip iprm counters**
- **show mpls ip iprm ldm**

For information about these commands, see the *Cisco IOS Debug Command Reference* and the *Cisco IOS MPLS Command Reference*.

Removal of Support for MPLS LSC and LC-ATM Features

The following MPLS LSC and LC-ATM features are no longer supported, starting with Cisco IOS Release 12.4(20)T:

- MPLS LSC
- LC-ATM
- MPLS Scalability Enhancements for LSC and ATM LSR
- MPLS LSC Redundancy
- MPLS--OAM Insertion and Loop Detection on LC-ATM
- MPLS CoS Multi-VC Mode for PA-A3
- MPLS over ATM: Virtual Circuit Merge
- MPLS Diff-Serv Aware Traffic Engineering over ATM
- VSI Master MIB

MPLS LSC and LC-ATM Configurations

Before upgrading to Cisco IOS Release 12.4(20)T, remove all the MPLS LSC and LC-ATM configurations from the routers in your network. If your core network has ATM links, you can use packet-based MPLS. See the MPLS Label Distribution Protocol Overview for more information. If you provide ATM access to customers, you can use the Any Transport over MPLS: ATM over MPLS feature. See Any Transport over MPLS for more information.

If you have MPLS LSC or LC-ATM features configured and you upgrade to Cisco IOS Release 12.4(20)T, the configuration is not accepted. The system displays "unrecognized command" errors for any commands that are no longer supported.

Removal of Support for MPLS LSC and LC-ATM Commands

The following commands are no longer supported, starting with Cisco IOS Release 12.4(20)T:

- **debug mpls atm-cos**
- **debug mpls atm-ldp api**
- **debug mpls atm-ldp failure**
- **debug mpls atm-ldp routes**
- **debug mpls atm-ldp states**
- **debug mpls xmpls cross-connect**
- **debug mpls xmpls errors**
- **debug mpls xmpls events**
- **debug mpls xmpls vc**
- **debug mpls xtagatm cross-connect**
- **debug mpls xtagatm errors**
- **debug mpls xtagatm events**
- **debug mpls xtagatm vc**
- **debug vsi api**
- **debug vsi errors**
- **debug vsi events**
- **debug vsi packets**
- **debug vsi param-groups**
- **extended-port**
- **interface xtagatm**
- **mpls atm control-vc**
- **mpls atm cos**

- **mpls atm disable-headend-vc**
- **mpls atm multi-vc**
- **mpls atm vpi**
- **mpls atm vp-tunnel**
- **mpls cos-map**
- **mpls ldp atm control-mode**
- **mpls ldp atm vc-merge**s
- **mpls prefix-map**
- **mpls request-labels for**
- **mpls traffic-eng atm cos global-pool**
- **mpls traffic-eng atm cos sub-pool**
- **show controllers vsi control-interface**
- **show controllers vsi descriptor**
- **show controllers vsi session**
- **show controllers vsi status**
- **show controllers vsi traffic**
- **show controllers xmpls**
- **show controllers xtagatm**
- **show interface xtagatm**
- **show mpls atm-ldp bindings**
- **show mpls atm-ldp bindwait**
- **show mpls atm-ldp capability**
- **show mpls atm-ldp summary**
- **show mpls cos-map**
- **show mpls prefix-map**
- **show xtagatm cos-bandwidth-allocation**
- **show xtagatm cross-connect**
- **show xtagatm vc**
- **snmp-server enable traps vsimaster**
- **tag-control-protocol vsi**

Additional References

Related Documents

Related Topic	Document Title
MPLS commands	*Cisco IOS MPLS Command Reference*
MPLS Label Distribution Protocol	MPLS Label Distribution Protocol Overview
Layer 2 VPN features over MPLS	Any Transport over MPLS

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

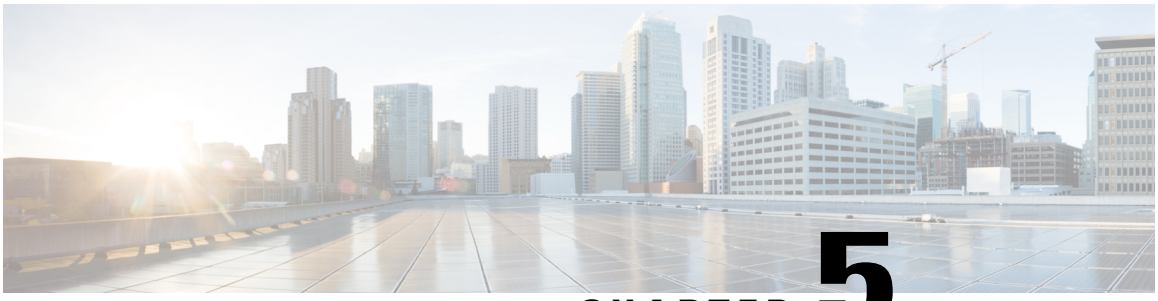
Feature Information for MPLS Infrastructure Changes

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 3: Feature Information for MPLS Infrastructure Changes

Feature Name	Releases	Feature Information
MPLS Infrastructure Changes	12.4(20)T Cisco IOS XE Release 3.5S	In Cisco IOS Release 12.4(20)T, this feature was introduced. In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.



CHAPTER 5

MPLS Static Labels

This document describes the Cisco MPLS Static Labels feature. The MPLS Static Labels feature provides the means to configure statically:

- The binding between a label and an IPv4 prefix
- The contents of an LFIB crossconnect entry

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.

Restrictions for MPLS Static Labels

- The trouble shooting process for MPLS static labels is complex.

- On a provider edge (PE) router for MPLS VPNs, there is no mechanism for statically binding a label to a customer network prefix (VPN IPv4 prefix).
- MPLS static crossconnect labels remain in the LFIB even if the router to which the entry points goes down.
- MPLS static crossconnect mappings remain in effect even with topology changes.
- MPLS static labels are not supported for label-controlled Asynchronous Transfer Mode (lc-atm).
- MPLS static bindings are not supported for local prefixes.

Prerequisites for MPLS Static Labels

The network must support the following Cisco IOS features before you enable MPLS static labels:

- Multiprotocol Label Switching (MPLS)
- Cisco Express Forwarding

Information About MPLS Static Labels

MPLS Static Labels Overview

Generally, label switching routers (LSRs) dynamically learn the labels they should use to label-switch packets by means of label distribution protocols that include:

- Label Distribution Protocol (LDP), the Internet Engineering Task Force (IETF) standard, used to bind labels to network addresses
- Resource Reservation Protocol (RSVP) used to distribute labels for traffic engineering (TE)
- Border Gateway Protocol (BGP) used to distribute labels for Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs)

To use a learned label to label-switch packets, an LSR installs the label into its Label Forwarding Information Base (LFIB).

The MPLS Static Labels feature provides the means to configure statically:

- The binding between a label and an IPv4 prefix
- The contents of an LFIB crossconnect entry

Benefits of MPLS Static Labels

Static Bindings Between Labels and IPv4 Prefixes

Static bindings between labels and IPv4 prefixes can be configured to support MPLS hop-by-hop forwarding through neighbor routers that do not implement LDP label distribution.

Static Crossconnects

Static crossconnects can be configured to support MPLS Label Switched Path (LSP) midpoints when neighbor routers do not implement either the LDP or RSVP label distribution, but do implement an MPLS forwarding path.

How to Configure MPLS Static Labels

Configuring MPLS Static Prefix Label Bindings

To configure MPLS static prefix/label bindings, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label range** *min-label max-label* [**static** *min-static-label max-static-label*]
4. **mpls static binding ipv4** *prefix mask* [**input**| **output** *nexthop*] label

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label range** *min-label max-label* [**static** *min-static-label max-static-label*] **Example:** `Router(config)# mpls label range 200 100000 static 16 199`	Specifies a range of labels for use with MPLS Static Labels feature. (Default is no labels reserved for static assignment.)
Step 4	**mpls static binding ipv4** *prefix mask* [**input**| **output** *nexthop*] label	Specifies static binding of labels to IPv4 prefixes. Bindings specified are installed automatically in the MPLS forwarding table as routing demands.

	Command or Action	Purpose
	Example: `Router(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 55`	

Verifying MPLS Static Prefix Label Bindings

To verify the configuration for MPLS static prefix/label bindings, use this procedure:

SUMMARY STEPS

1. Enter **show mpls label range** command. The output shows that the new label ranges do not take effect until a reload occurs:
2. Enter the **show mpls static binding ipv4** command to show the configured static prefix/label bindings:
3. Use the **show mpls forwarding-table** command to determine which static prefix/label bindings are currently in use for MPLS forwarding.

DETAILED STEPS

Step 1 Enter **show mpls label range** command. The output shows that the new label ranges do not take effect until a reload occurs:

Example:

```
Router# show mpls label range

Downstream label pool: Min/Max label: 16/100000
   [Configured range for next reload: Min/Max label: 200/100000]
Range for static labels: Min/Max/Number: 16/199
```

The following output from the **show mpls label range** command, executed after a reload, indicates that the new label ranges are in effect:

Example:

```
Router# show mpls label range

Downstream label pool: Min/Max label: 200/100000
Range for static labels: Min/Max/Number: 16/199
```

Step 2 Enter the **show mpls static binding ipv4** command to show the configured static prefix/label bindings:

Example:

```
Router# show mpls static binding ipv4
10.17.17.17/32: Incoming label: 251 (in LIB)
  Outgoing labels:
```

```
     10.0.0.1                  18
10.18.18.18/32: Incoming label: 201 (in LIB)
  Outgoing labels:
10.0.0.1 implicit-null
```

Step 3 Use the **show mpls forwarding-table** command to determine which static prefix/label bindings are currently in use for MPLS forwarding.

Example:

```
Router# show mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
201    Pop tag     10.18.18.18/32   0          PO1/1/0    point2point
       2/35        10.18.18.18/32   0          AT4/1/0.1  point2point
251    18          10.17.17.17/32   0          PO1/1/0    point2point
```

Configuring MPLS Static Crossconnects

To configure MPLS static crossconnects, use the following command beginning in global configuration mode:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label range** *min-label max-label* [**static** *min-static-label max-static-label*]
4. **mpls static binding ipv4** *prefix mask* [**input**| **output** *nexthop*] *label*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label range** *min-label max-label* [**static** *min-static-label max-static-label*]	Specifies a range of labels for use with MPLS Static Labels feature.

	Command or Action	Purpose
	Example: `Router(config)# mpls label range 200 100000 static 16 199`	(Default is no labels reserved for static assignment.)
Step 4	**mpls static binding ipv4** *prefix mask* [**input**| **output** *nexthop*] *label* **Example:** `Router(config)# Router(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 55`	Specifies static binding of labels to IPv4 prefixes. Bindings specified are installed automatically in the MPLS forwarding table as routing demands.

Verifying MPLS Static Crossconnect Configuration

To verify the configuration for MPLS static crossconnects, use this procedure:

SUMMARY STEPS

1. Use the **show mpls static crossconnect** command to display information about crossconnects that have been configured:

DETAILED STEPS

Use the **show mpls static crossconnect** command to display information about crossconnects that have been configured:

Example:

```
Router# show mpls static crossconnect
Local  Outgoing    Outgoing   Next Hop
label  label       interface
34     22          pos3/0/0   point2point  (in LFIB)
```

Monitoring and Maintaining MPLS Static Labels

To monitor and maintain MPLS static labels, use one or more of the following commands:

SUMMARY STEPS

1. **enable**
2. **show mpls forwarding-table**
3. **show mpls label range**
4. **show mpls static binding ipv4**
5. **show mpls static crossconnect**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls forwarding-table** **Example:** `Router# show mpls forwarding-table`	Displays the contents of the MPLS LFIB.
Step 3	**show mpls label range** **Example:** `Router# show mpls label range`	Displays information about the static label range.
Step 4	**show mpls static binding ipv4** **Example:** `Router# show mpls static binding ipv4`	Displays information about the configured static prefix/label bindings.
Step 5	**show mpls static crossconnect** **Example:** `Router# show mpls static crossconnect`	Displays information about the configured crossconnects.

Configuration Examples for MPLS Static Labels

Example Configuring MPLS Static Prefixes Labels

In the following output, the **mpls label range** command reconfigures the range used for dynamically assigned labels from 16 to 100000 to 200 to 100000 and configures a static label range of 16 to 199.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mpls label range 200 100000 static 16 199
% Label range changes take effect at the next reload.
Router(config)# end
```

In the following output, the **show mpls label range** command indicates that the new label ranges do not take effect until a reload occurs:

```
Router# show mpls label range

Downstream label pool: Min/Max label: 16/100000
   [Configured range for next reload: Min/Max label: 200/100000]
Range for static labels: Min/Max/Number: 16/199
```

In the following output, the **show mpls label range** command, executed after a reload, indicates that the new label ranges are in effect:

```
Router# show mpls label range

Downstream label pool: Min/Max label: 200/100000
Range for static labels: Min/Max/Number: 16/199
```

In the following output, the **mpls static binding ipv4** commands configure static prefix/label bindings. They also configure input (local) and output (remote) labels for various prefixes:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 55
Router(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 output 10.0.0.66 2607
Router(config)# mpls static binding ipv4 10.6.0.0 255.255.0.0 input 17
Router(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 output 10.13.0.8 explicit-null
Router(config)# end
```

In the following output, the **show mpls static binding ipv4** command displays the configured static prefix/label bindings:

```
Router# show mpls static binding ipv4

10.0.0.0/8: Incoming label: none;
  Outgoing labels:
10.13.0.8          explicit-null
10.0.0.0/8: Incoming label: 55 (in LIB)
  Outgoing labels:
     10.0.0.66             2607
10.66.0.0/16: Incoming label: 17 (in LIB)
  Outgoing labels:  None
```

Example Configuring MPLS Static Crossconnects

In the following output, the **mpls static crossconnect** command configures a crossconnect from incoming label 34 to outgoing label 22 out interface pos3/0/0:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mpls static crossconnect 34 pos3/0/0 22
Router(config)# end
```

In the following output, the **show mpls static crossconnect** command displays the configured crossconnect:

```
Router# show mpls static crossconnect
Local  Outgoing    Outgoing   Next Hop
label  label       interface
34     22          pos3/0/0   point2point  (in LFIB)
```

Additional References

Related Documents

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

Standards

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

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	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
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

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 MPLS Static Labels

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 4: Feature Information for MPLS Static Labels

Feature Name	Releases	Feature Information
MPLS Static Labels	Cisco IOS XE Release 2.1	The MPLS Static Labels feature provides the means to configure the following items statically: • The binding between a label and an IPv4 prefix • The contents of an LFIB crossconnect entry The following commands were introduced or modified: **debug mpls static binding**, **mpls label range**, **mpls static binding ipv4**, **mpls static crossconnect**, **show mpls label range**, **show mpls static binding ipv4**, **show mpls static crossconnect**

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.

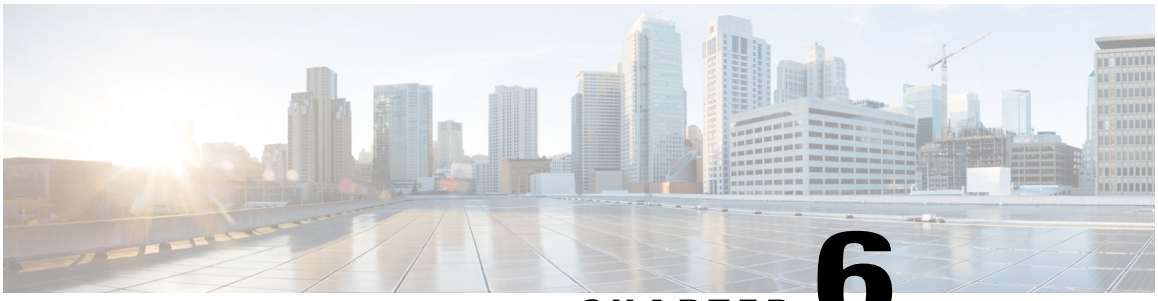
Routing Information Base --See RIB.

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 6

MPLS Multilink PPP Support

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.

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.

Prerequisites for MPLS Multilink PPP Support

- Cisco Express Forwarding must be enabled.
- Multiprotocol Label Switching (MPLS) must be enabled on provider edge (PE) and provider (P) devices.
- Cisco Express Forwarding switching must be enabled on the interface by using the **ip route-cache cef** command.

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 5: 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

MPLS L3 VPN Feature	CE-to-PE Links	PE-to-P Links	CSC CE-to-PE Links
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
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 6: 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

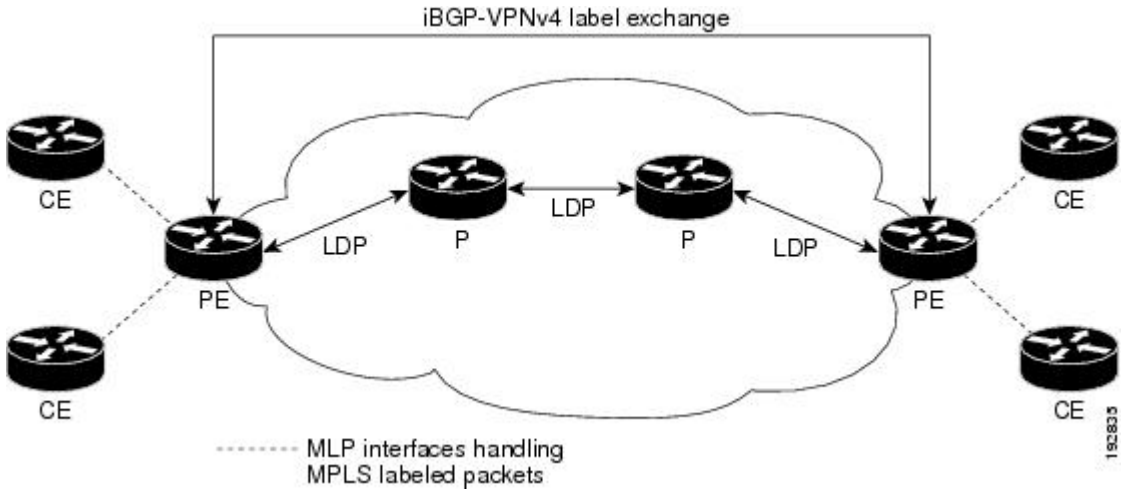
MPLS QoS Feature	CE-to-PE Links	PE-to-P Links	CSC CE-to-PE Links
Policer with EXP bit-marking using MQC-3 action	Supported	Supported	Supported
Support for EXP bits in MPLS accounting	Supported	Supported	Supported

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



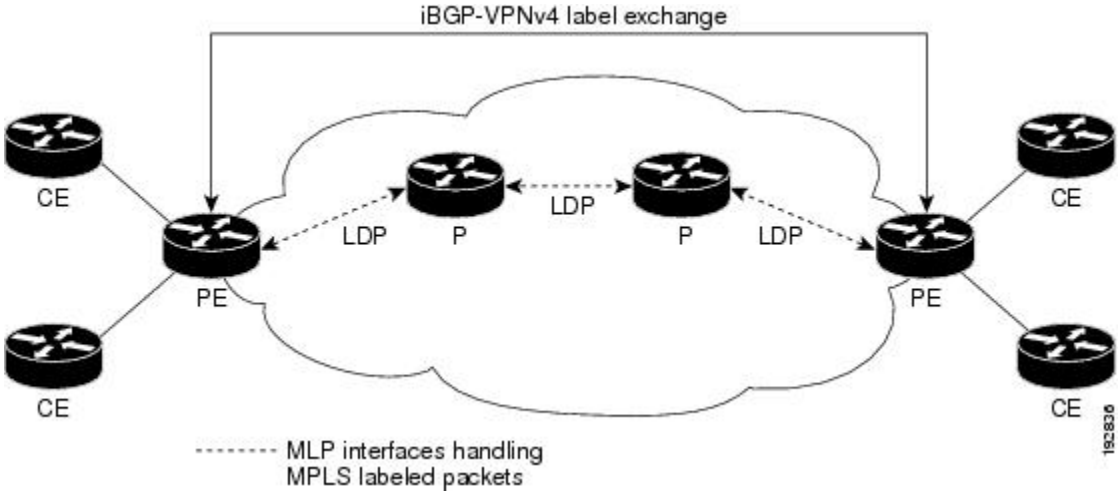
The PE-to-CE routing protocols that are supported for the MPLS Multilink PPP Support feature are external BGP (eBGP), Open Shortest Path First (OSPF), and Enhanced Interior Gateway Routing Protocol (EIGRP). Static routes are also supported between the CE and PE device.

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), header compression, 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



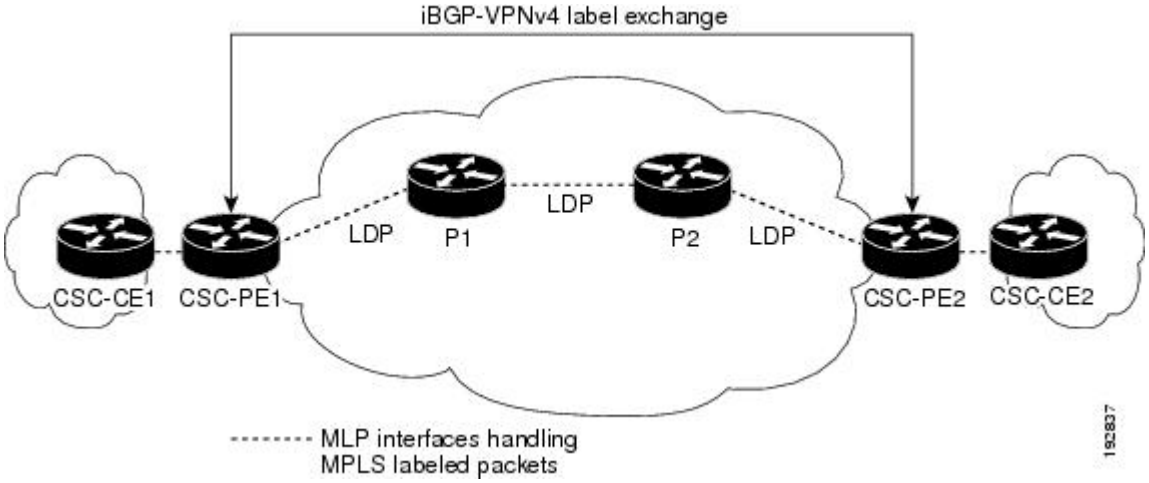
You employ MLP in the PE-to-P or P-to-P links primarily so that you can reduce the number of Interior Gateway Protocol (IGP) adjacencies and facilitate the load sharing of traffic.

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

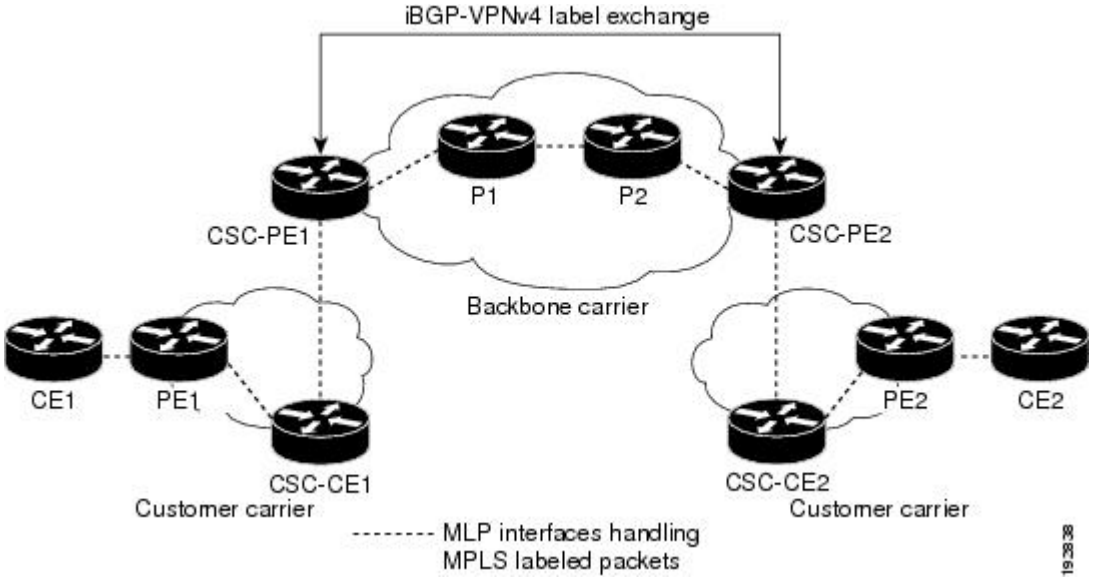
The figure below shows a typical Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) Carrier Supporting Carrier (CSC) network where Multilink PPP (MLP) is configured on the CSC customer edge (CE)-to-provider edge (PE) links.

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.

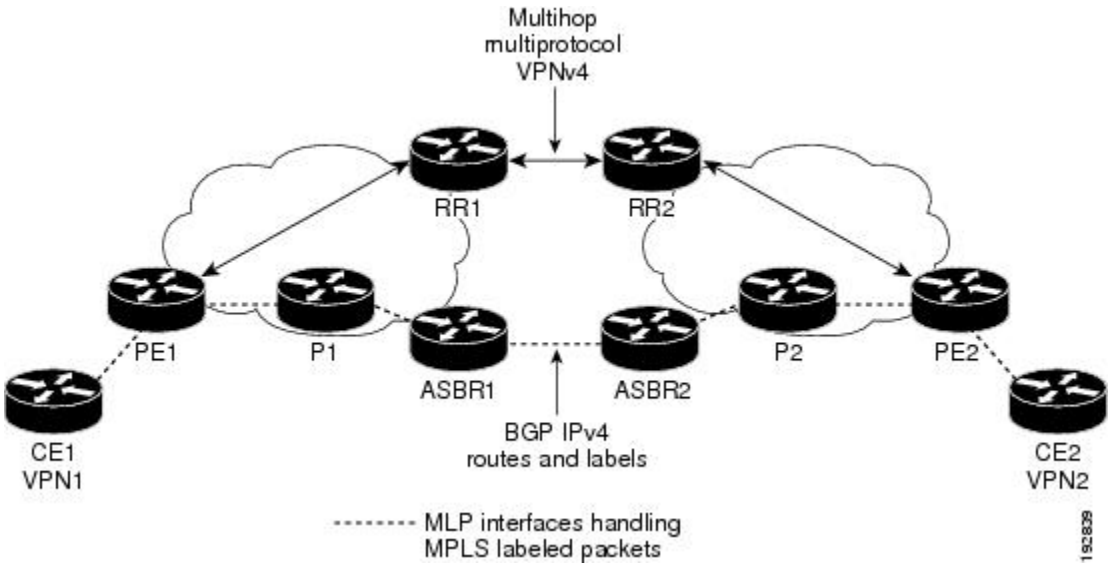
Figure 4: MLP Supported Links with MPLS VPN Carrier Supporting Carrier



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.

Enabling Cisco Express Forwarding

Perform the following task to enable Cisco Express Forwarding. Cisco Express Forwarding is required for the forwarding of MLP traffic.

Before You Begin

Multilink PPP requires the configuration of Cisco Express Forwarding. To find out if Cisco Express Forwarding is enabled on your device, enter the**show ip cef** command. If Cisco Express Forwarding is enabled, you receive output that looks like the following:

```
Device# show ip cef
Prefix              Next Hop            Interface
```

```
10.2.61.8/24        192.168.100.1       FastEthernet1/0/0
                    192.168.101.1       FastEthernet6/1/0
```

If Cisco Express Forwarding is not enabled on your platform, the output for the **show ip cef** command looks like the following:

```
Device# show ip cef
%CEF not running
```

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef**
4. **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	**ip cef** **Example:** `Device(config)# ip cef`	Enables Cisco Express Forwarding.
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.

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. **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.

	Command or Action	Purpose
Step 6	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables MLP on an interface.
Step 7	**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** *range*
5. **exit**
6. **interface serial** *slot*/*subslot*/*port*[**.***subinterface*]
7. **ip route-cache** [**cef**]
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.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**controller** {**t1** | **e1**} *slot*/*port* **Example:** `Device# controller t1 1/3`	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 hardware installation manual for the specific slot numbers and port numbers.
Step 4	**channel-group** *channel-number* **timeslots** *range* **Example:** `Device(config-controller)# channel-group 1 timeslots 1`	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 0 to 23. When an E1 data line is configured, channel-group numbers can be values from 0 to 30. • The **timeslots** *range* keyword and argument specifies one or more time slots or ranges of time slots belonging to the channel group. The first time slot is numbered 1. For a T1 controller, the time slot range is from 1 to 24. For an E1 controller, the time slot range is from 1 to 31. You can specify a time slot range (for example, 1-29), individual time slots separated by commas (for example 1, 3, 5), or a combination of the two (for example 1-14, 15, 17-31).
Step 5	**exit** **Example:** `Device(config-controller)# exit`	Returns to global configuration mode.
Step 6	**interface serial** *slot*/*subslot*/*port*[**.***subinterface*] **Example:** `Device(config)# interface serial 1/0/0:1`	Configures a serial interface and enters interface configuration mode.
Step 7	**ip route-cache** [**cef**] **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.

	Command or Action	Purpose
Step 8	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes any specified 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.

	Command or Action	Purpose
Step 14	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Disabling PPP Multilink Fragmentation

Perform this task to disable PPP multilink fragmentation. PPP multilink fragmentation is enabled by default.

Enabling fragmentation reduces the delay latency among bundle links, but adds some load to the CPU. Disabling fragmentation might produce better throughput.

If your data traffic is consistently of a similar size, we recommend disabling fragmentation. In this case, the benefits of fragmentation can be outweighed by the added load on the CPU.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ppp multilink fragmentation disable**
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 number* **Example:** `Device(config)# interface serial 1/0/0`	Configures an interface type and enters interface configuration mode. • The *type* argument indicates the type of interface to be configured.

	Command or Action	Purpose
		• The *number* argument specifies the port, connector, or interface card number. The numbers are assigned at the factory at the time of installation or when the interface is added to a system, and they can be displayed with the **show interfaces** command.
Step 4	**ppp multilink fragmentation disable** **Example:** `Device(config-if)# ppp multilink fragmentation disable`	Disables packet fragmentation.
Step 5	**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:

Step 3 **show ppp multilink**
Verifies that you have created a multilink bundle.

Example:

Step 4 **show ppp multilink interface** *interface-bundle*
Displays information about a specific MLP interface.

Example:

Step 5 **show interface** *type number*
Displays information about serial interfaces in your configuration.

Example:

```
Device#


  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#


  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

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
!


!

 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

!
!
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: Enabling Cisco Express Forwarding

The following example shows how to enable Cisco Express Forwarding for Multilink PPP (MLP) configurations:

```
Device> enable
Device# configure terminal
Device(config)# ip cef
```

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
```

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



 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
```

Additional References for MPLS Multilink PPP Support

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
Basic MPLS VPNs	"MPLS Virtual Private Networks" chapter in the *MPLS Layer 3 VPNs Configuration Guide*

RFCs

RFCs	Title
RFC 1990	*The PPP Multilink Protocol (MP)*

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 MPLS Multilink PPP 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 7: Feature Information for MPLS Multilink PPP Support

Feature Name	Releases	Feature Information
MPLS Multilink PPP Support	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.9S	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). In Cisco IOS XE Release 2.1, support was added for the Cisco ASR 1000 Series Routers. In Cisco IOS XE Release 3.9S, support was added for the Cisco ASR 903 Router.

Glossary

bundle—A group of interfaces connected by parallel links between two systems that have agreed to use Multilink PPP (MLP) over those links.

CBWFQ—class-based weighted fair queueing. A queueing option that extends the standard Weighted Fair Queueing (WFQ) functionality to provide support for user-defined traffic classes.

Cisco Express Forwarding—A proprietary form of switching that optimizes network performance and scalability for networks with large and dynamic traffic patterns, such as the Internet, and for networks characterized by intensive web-based applications or interactive sessions. Although you can use Cisco Express Forwarding in any part of a network, it is designed for high-performance, highly resilient Layer 3 IP backbone switching.

EIGRP—Enhanced Interior Gateway Routing Protocol. An advanced version of the Interior Gateway Routing Protocol (IGRP) developed by Cisco. It provides superior convergence properties and operating efficiency, and combines the advantages of link-state protocols with those of distance vector protocols.

IGP—Interior Gateway Protocol. An Internet protocol used to exchange routing information within an autonomous system. Examples of common Internet IGPs include Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

IGRP—Interior Gateway Routing Protocol. An Interior Gateway Protocol (IGP) developed by Cisco to address the issues associated with routing in large, heterogeneous networks. Compare with Enhanced Interior Gateway Routing Protocol (EIGRP).

IS-IS—Intermediate System-to-Intermediate System. An Open Systems Interconnection (OSI) link-state hierarchical routing protocol, based on DECnet Phase V routing, in which IS-IS devices exchange routing information based on a single metric to determine network topology.

LCP—Link Control Protocol. A protocol that establishes, configures, and tests data link connections for use by PPP.

LFI—ink fragmentation and interleaving. The LFI feature reduces delay on slower-speed links by breaking up large datagrams and interleaving low-delay traffic packets with the smaller packets resulting from the fragmented datagram. LFI allows reserve queues to be set up so that Real-Time Protocol (RTP) streams can be mapped into a higher priority queue in the configured weighted fair queue set.

link—One of the interfaces in a bundle.

LLQ—low latency queueing. A quality of service QoS queueing feature that provides a strict priority queue (PQ) for voice traffic and weighted fair queues for other classes of traffic. It is also called priority queueing/class-based weighted fair queueing (PQ/CBWFQ).

MLP—Multilink PPP. A method of splitting, recombining, and sequencing datagrams across multiple logical links. The use of MLP increases throughput between two sites by grouping interfaces and then load balancing packets over the grouped interfaces (called a bundle). Splitting packets at one end, sending them over the bundled interfaces, and recombining them at the other end achieves load balancing.

MQC—Modular QoS CLI. MQC is a CLI structure that allows users to create traffic polices and attach these polices to interfaces. MQC allows users to specify a traffic class independently of QoS policies.

NCP—Network Control Protocol. A series of protocols for establishing and configuring different network layer protocols (such as for AppleTalk) over PPP.

OSPF—Open Shortest Path First. A link-state, hierarchical Interior Gateway Protocol (IGP) routing algorithm proposed as a successor to Routing Information Protocol (RIP) in the Internet community. OSPF features

include least-cost routing, multipath routing, and load balancing. OSPF was derived from an early version of the IS-IS protocol.

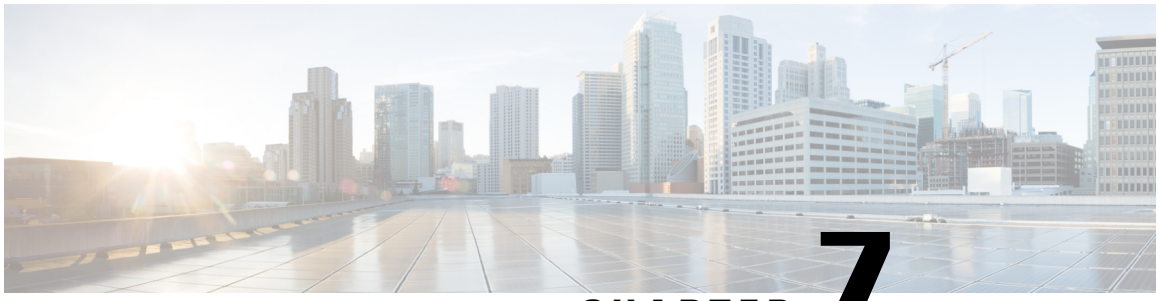
PPP—Point-to-Point Protocol. A successor to the Serial Line Interface Protocol (SLIP) that provides device-to-device and host-to-network connections over synchronous and asynchronous circuits. PPP works with several network layer protocols (such as IP, Internetwork Packet Exchange [IPX], and AppleTalk Remote Access [ARA]). PPP also has built-in security mechanisms (such as Challenge Handshake Authentication Protocol [CHAP] and Password Authentication Protocol [PAP]). PPP relies on two protocols: Link Control Protocol (LCP) and Network Control Protocol (NCP).

RIP—Routing Information Protocol. A version of Interior Gateway Protocol (IGP) that is supplied with UNIX Berkeley Standard Distribution (BSD) systems. Routing Information Protocol (RIP) is the most common IGP in the Internet. It uses hop count as a routing metric.

Virtual bundle interface—An interface that represents the master link of a bundle. It is not tied to any physical interface. Data going over the bundle is transmitted and received through the master link.

WFQ—weighted fair queueing. A congestion management algorithm that identifies conversations (in the form of traffic streams), separates packets that belong to each conversation, and ensures that capacity is shared fairly among the individual conversations. WFQ is an automatic way of stabilizing network behavior during congestion and results in improved performance and reduced retransmission.

WRED—weighted random early detection. A queueing method that ensures that high-precedence traffic has lower loss rates than other traffic during times of congestion.



CHAPTER 7

6PE Multipath

The 6PE multipath feature uses multiprotocol internal BGP (MP-iBGP) to distribute IPv6 routes over the MPLS IPv4 core network and to attach an MPLS label to each route

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 6PE Multipath

6PE Multipath

Internal and external BGP multipath for IPv6 allows the IPv6 device to load balance between several paths (for example, the same neighboring autonomous system or subautonomous system, or the same metric) to reach its destination. The 6PE multipath feature uses MP-iBGP to distribute IPv6 routes over the MPLS IPv4 core network and to attach an MPLS label to each route.

When MP-iBGP multipath is enabled on the 6PE device, all labeled paths are installed in the forwarding table with MPLS information (label stack) when MPLS information is available. This functionality enables 6PE to perform load balancing.

How to Configure 6PE Multipath

Configuring IBGP Multipath Load Sharing

Perform this task to configure IBGP multipath load sharing and control the maximum number of parallel IBGP routes that can be installed in a routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**unicast**]
5. **maximum-paths ibgp** *number-of-paths*

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	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family ipv6** [**unicast**] **Example:** `Device(config-router)# address-family ipv6`	Specifies the IPv6 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if the **unicast** keyword is not specified with the **address-family ipv6** command.

	Command or Action	Purpose
Step 5	**maximum-paths ibgp** *number-of-paths* **Example:** `Device(config-router)# maximum-paths ibgp 3`	Controls the maximum number of parallel IBGP routes that can be installed in a routing table.

Configuration Examples for 6PE Multipath

Example: Configuring 6PE Multipath

```
Device# show ipv6 cef internals
IPv6 CEF is enabled and running
Slow processing intvl = 1 seconds backoff level current/max 0/0
0 unresolved prefixes, 0 requiring adjacency update
IPv6 CEF default table
14 prefixes tableid 0
table version 17
root 6283F5D0
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Master Commands List, All Releases
IPv6 commands	IPv6 Command Reference
Cisco IOS IPv6 features	IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

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

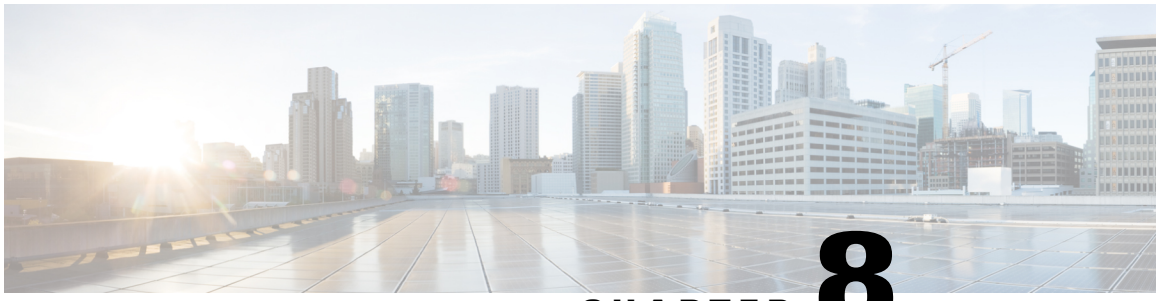
Feature Information for 6PE Multipath

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 8: Feature Information for 6PE Multipath

Feature Name	Releases	Feature Information
6PE Multipath	Cisco IOS XE Release 3.1S	The 6PE multipath feature uses MP-iBGP to distribute IPv6 routes over the MPLS IPv4 core network and to attach an MPLS label to each route. The following commands were introduced or modified: **maximum-paths ibgp**, **router bgp**, **show ipv6 cef internals**.



CHAPTER 8

IPv6 Switching: Provider Edge Router over MPLS

Multiprotocol Label Switching (MPLS) is deployed by many service providers in their IPv4 networks. Service providers want to introduce IPv6 services to their customers, but changes to their existing IPv4 infrastructure can be expensive and the cost benefit for a small amount of IPv6 traffic does not make economic sense. Several integration scenarios have been developed to leverage an existing IPv4 MPLS infrastructure and add IPv6 services without requiring any changes to the network backbone. This document describes how to implement IPv6 over MPLS.

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.

Prerequisites for IPv6 Switching: Provider Edge Router over MPLS

Before the IPv6 Provider Edge Router over MPLS (6PE) feature can be implemented, MPLS must be running over the core IPv4 network. If Cisco devices are used, Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled for both IPv4 and IPv6 protocols. This module assumes that you are familiar with MPLS.

Information About IPv6 Switching: Provider Edge Router over MPLS

Benefits of Deploying IPv6 over MPLS Backbones

IPv6 over MPLS backbones enables isolated IPv6 domains to communicate with each other over an MPLS IPv4 core network. This implementation requires only a few backbone infrastructure upgrades and no reconfiguration of core devices because forwarding is based on labels rather than the IP header itself, providing a very cost-effective strategy for the deployment of IPv6.

Additionally, the inherent Virtual Private Network (VPN) and MPLS traffic engineering (MPLS-TE) services available within an MPLS environment allow IPv6 networks to be combined into IPv4 VPNs or extranets over an infrastructure supporting IPv4 VPNs and MPLS-TE.

IPv6 on the Provider Edge Devices

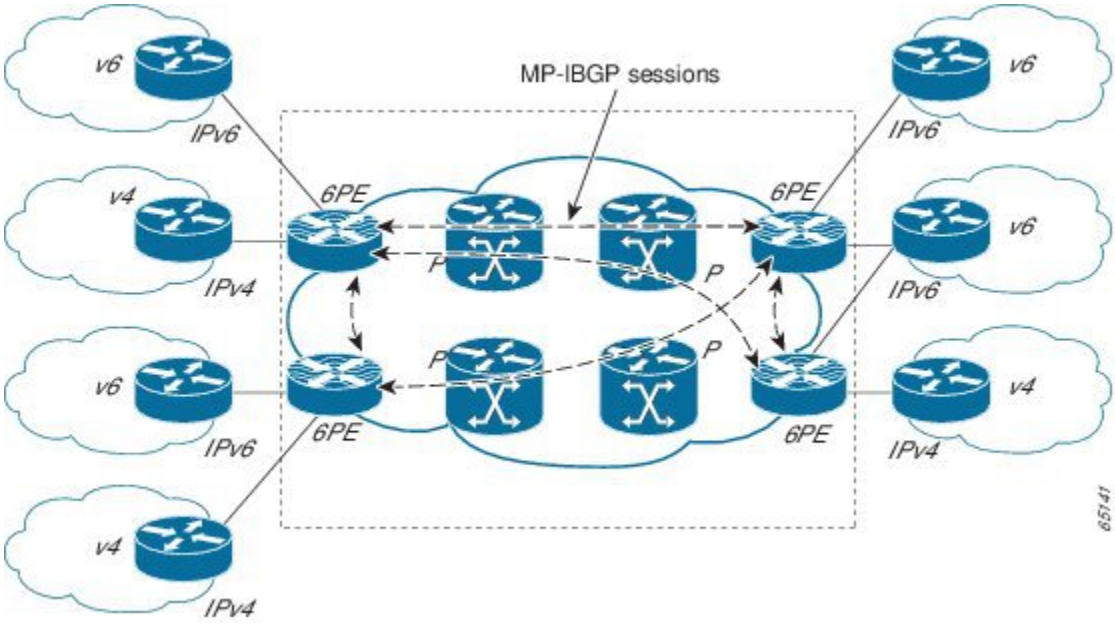
The Cisco implementation of IPv6 Provider Edge Router over MPLS is called 6PE, and it enables IPv6 sites to communicate with each other over an MPLS IPv4 core network using MPLS label switched paths (LSPs). This feature relies on multiprotocol Border Gateway Protocol (BGP) extensions in the IPv4 network configuration on the provider edge (PE) device to exchange IPv6 reachability information in addition to an MPLS label for each IPv6 address prefix to be advertised. Edge devices are configured to be dual stack running both IPv4 and IPv6, and use the IPv4 mapped IPv6 address for IPv6 prefix reachability exchange.

A hierarchy of labels is imposed on the 6PE ingress device to keep the IPv6 traffic transparent to all the core devices. The top label provides connectivity inside the IPv4 MPLS core network and the label is distributed by Label Distribution Protocol (LDP), Tag Distribution Protocol (TDP), or Resource Reservation Protocol (RSVP). TDP and LDP can both be used for label distribution, but RSVP is used only in the context of MPLS-TE label exchange. The bottom label, automatically assigned to the IPv6 prefix of the destination, is distributed by multiprotocol BGP and used at each 6PE egress device for IPv6 forwarding.

In the figure below the 6PE devices are configured as dual stack devices able to route both IPv4 and IPv6 traffic. Each 6PE device is configured to run LDP, TDP, or RSVP (if traffic engineering is configured) to bind the IPv4 labels. The 6PE devices use multiprotocol BGP to exchange reachability information with the other 6PE devices within the MPLS domain, and to distribute IPv6 labels between them. All 6PE and core devices--P devices in Figure 3--within the MPLS domain share a common IPv4 Interior Gateway Protocol

(IGP) such as Open Shortest Path First (OSPF) or Integrated Intermediate System-to-Intermediate System (IS-IS).

Figure 6: 6PE Device Topology



The interfaces on the 6PE devices connecting to the CE device can be configured to forward IPv6 traffic, IPv4 traffic, or both types of traffic depending on the customer requirements. 6PE devices advertise IPv6 reachability information learned from their 6PE peers over the MPLS cloud. Service providers can delegate an IPv6 prefix from their registered IPv6 prefixes over the 6PE infrastructure; otherwise, there is no impact on the CE device.

The P devices in the core of the network are not aware that they are switching IPv6 packets. Core devices are configured to support MPLS and the same IPv4 IGP as the PE devices to establish internal reachability inside the MPLS cloud. Core devices also use LDP, TDP, or RSVP for binding IPv4 labels. Implementing the Cisco 6PE feature does not have any impact on the MPLS core devices.

Within the MPLS network, IPv6 traffic is forwarded using label switching, making the IPv6 traffic transparent to the core of the MPLS network. No IPv6 over IPv4 tunnels or Layer 2 encapsulation methods are required.

How to Deploy IPv6 Switching: Provider Edge Router over MPLS

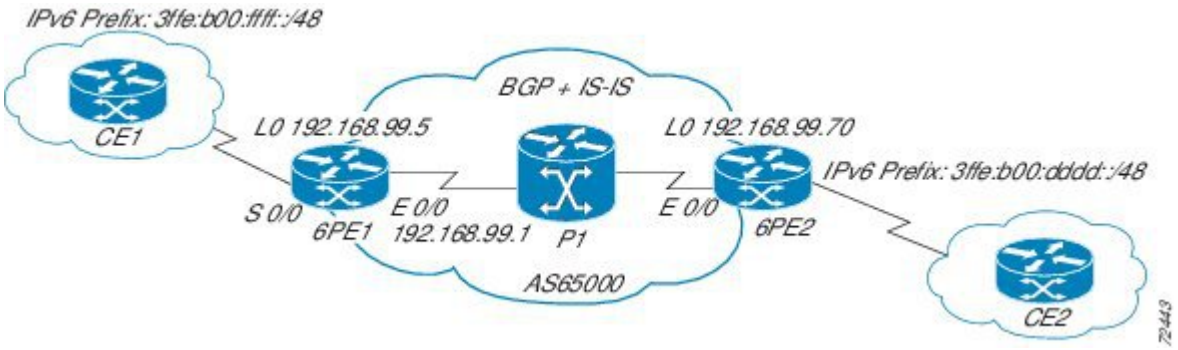
Deploying IPv6 on the Provider Edge Devices (6PE)

Specifying the Source Address Interface on a 6PE Device

Two configuration tasks using the network shown in the figure below are required at the 6PE1 device to enable the 6PE feature.

The customer edge device--CE1 in the figure below--is configured to forward its IPv6 traffic to the 6PE1 device. The P1 device in the core of the network is assumed to be running MPLS, a label distribution protocol, an IPv4 IGP, and Cisco Express Forwarding or distributed Cisco Express Forwarding, and does not require any new configuration to enable the 6PE feature.

Figure 7: 6PE Configuration Example



Before You Begin

- The 6PE devices--the 6PE1 and 6PE2 devices in the figure below--must be members of the core IPv4 network. The 6PE device interfaces attached to the core network must be running MPLS, the same label distribution protocol, and the same IPv4 IGP, as in the core network.
- The 6PE devices must also be configured to be dual stack to run both IPv4 and IPv6.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **ipv6 cef**
5. **interface** *type number*
6. **ipv6 address** *ipv6-address / prefix-length | prefix-name sub-bits / prefix-length*

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	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**ipv6 cef** **Example:** `Device(config)# ipv6 cef`	Enables IPv6 Cisco Express Forwarding.
Step 5	**interface** *type number* **Example:** `Device(config)# interface`	Specifies an interface type and number and enters interface configuration mode. • In the context of this feature, the interface to be configured is the interface communicating with the CE device.
Step 6	**ipv6 address** *ipv6-address / prefix-length | prefix-name sub-bits / prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:FFFF::2/64`	Configures an IPv6 address based on an IPv6 general prefix and enable IPv6 processing on an interface.

Binding and Advertising the 6PE Label to Advertise Prefixes

Perform this task to enable the binding and advertising of labels when advertising IPv6 prefixes to a specified BGP neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** { *ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
7. **address-family ipv6** [**unicast**]
8. **neighbor** {*ip-address* | *peer-group-name*| *ipv6-address*} **activate**
9. **neighbor** {*ip-address* | *ipv6-address*} **send-label**

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	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 4	**no bgp default ipv4-unicast** **Example:** `Device(config-router)# no bgp default ipv4-unicast`	Disables the IPv4 unicast address family for the BGP routing process specified in the previous step. **Note** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the **neighbor remote-as** command unless you configure the **no bgp default ipv4-unicast**command before configuring the **neighbor remote-as** command.
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.99.70 remote-as 65000`	Adds the IP address of the neighbor in the specified autonomous system to the BGP neighbor table of the local device.

	Command or Action	Purpose
Step 6	**neighbor** { *ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 192.168.99.70 update-source Loopback 0`	Specifies the interface whose IPv4 address is to be used as the source address for the peering. • In the context of this task, the interface must have an IPv4 address with a 32-bit mask configured. Use of a loopback interface is recommended. This address is used to determine the IPv6 next hop by the peer 6PE.
Step 7	**address-family ipv6** [**unicast**] **Example:** `Device(config-router)# address-family ipv6`	Specifies the IPv6 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if the **unicast** keyword is not specified with the **address-family ipv6** command.
Step 8	**neighbor** {*ip-address* | *peer-group-name*| *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.99.70 activate`	Enables the neighbor to exchange prefixes for the IPv6 address family with the local device.
Step 9	**neighbor** {*ip-address* | *ipv6-address*} **send-label** **Example:** `Device(config-router-af)# neighbor 192.168.99.70 send-label`	Advertises the capability of the device to send MPLS labels with BGP routes. • In IPv6 address family configuration mode this command enables binding and advertisement of labels when advertising IPv6 prefixes in BGP.

Configuring IBGP Multipath Load Sharing

Perform this task to configure IBGP multipath load sharing and control the maximum number of parallel IBGP routes that can be installed in a routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**unicast**]
5. **maximum-paths ibgp** *number-of-paths*

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	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family ipv6** [**unicast**] **Example:** `Device(config-router)# address-family ipv6`	Specifies the IPv6 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if the **unicast** keyword is not specified with the **address-family ipv6** command.
Step 5	**maximum-paths ibgp** *number-of-paths* **Example:** `Device(config-router)# maximum-paths ibgp 3`	Controls the maximum number of parallel IBGP routes that can be installed in a routing table.

Configuration Examples for IPv6 Switching: Provider Edge Router over MPLS

Example: Provider Edge Device

The 6PE device is configured for both IPv4 and IPv6 traffic. Gigabit Ethernet interface 0/0/0 is configured with an IPv4 address and is connected to a device in the core of the network. Integrated IS-IS and TDP configurations on this device are similar to the P1 device.

Device 6PE1 exchanges IPv6 routing information with another 6PE device using internal BGP (IBGP) established over an IPv4 connection so that all the **neighbor** commands use the IPv4 address of the 6PE2 device. All the BGP peers are within autonomous system 65000, so synchronization with IGP is turned off for IPv4. In IPv6 address family configuration mode, synchronization is disabled by default.

IPv6 and Cisco Express Forwarding for IPv6 are enabled, the 6PE2 neighbor is activated, and label binding and advertisement is enabled for IPv6 prefixes using the **neighbor send-label** command. Connected and static IPV6 routes are redistributed using BGP. If IPv6 packets are generated in the local device, the IPv6 address for MPLS processing will be the address of loopback interface 0.

In the following example, serial interface 0/0 connects to the customer and the IPv6 prefix delegated to the customer is 2001:DB8:ffff::/48, which is determined from the service provider IPv6 prefix. A static route is configured to route IPv6 packets between the 6PE route and the CE device.

```
ip cef
ipv6 cef
ipv6 unicast-routing
!
mpls ipv6 source-interface Loopback0
tag-switching tdp router-id Loopback0
!
interface Loopback0
 ip address 192.168.99.5 255.255.255.255
 ipv6 address 2001:DB8:1000:1::1/64
!
interface GigabitEthernet0/0/0
 description to_P_router
 ip address 192.168.99.1 255.255.255.252
 ip router isis
 tag-switching ip
!
interface GigabitEthernet0/1/0
 description to_CE_router
 no ip address
 ipv6 address 2001:DB8:FFFF::1/64
!
router isis
 passive-interface Loopback0
 net 49.0001.1921.6809.9005.00
!
router bgp 65000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.99.70 remote-as 65000
 neighbor 192.168.99.70 description to_6PE2
 neighbor 192.168.99.70 update-source Loopback0
 !
 address-family ipv6
 neighbor 192.168.99.70 activate
 neighbor 192.168.99.70 send-label
 network 2001:DB8:FFFF::/48
 exit-address-family
!
ipv6 route 2001:DB8:FFFF::/48 GigabitEthernet0/0/0 2001:DB8:FFFF::2
```

Example: Core Device

In the following example, the device in the core of the network is running MPLS, IS-IS, and IPv4 only. The Gigabit Ethernet interfaces are configured with IPv4 address and are connected to the 6PE devices. IS-IS is the IGP for this network and the P1 and 6PE devices are in the same IS-IS area 49.0001. Tag Distribution

Protocol (TDP) and tag switching are enabled on both the Gigabit Ethernet interfaces. Cisco Express Forwarding is enabled in global configuration mode.

```
ip cef
!
tag-switching tdp router-id Loopback0
!
interface Loopback0
 ip address 192.168.99.200 255.255.255.255
!
interface GigabitEthernet0/0/0
 description to_6PE1
 ip address 192.168.99.2 255.255.255.252
 ip router isis
 tag-switching ip
!
interface GigabitEthernet0/1/0
 description to_6PE2
 ip address 192.168.99.66 255.255.255.252
 ip router isis
 tag-switching ip
router isis
 passive-interface Loopback0
 net 49.0001.1921.6809.9200.00
```

Example: Monitoring 6PE

In the following example, output information about an IPv6 route is displayed using the **show bgp ipv6** command with an IPv6 prefix:

```
Device# show bgp ipv6 2001:DB8:DDDD::/48

BGP routing table entry for 2001:DB8:DDDD::/48, version 15
Paths: (1 available, best #1, table Global-IPv6-Table)
  Not advertised to any peer
  Local
    ::FFFF:192.168.99.70 (metric 20) from 192.168.99.70 (192.168.99.70)
      Origin IGP, localpref 100, valid, internal, best
```

In the following example, output information about a BGP peer including the IPv6 label capability is displayed using the **show bgp ipv6 neighbors** command with an IP address:

```
Device# show bgp ipv6 neighbors 192.168.99.70

BGP neighbor is 192.168.99.70, remote AS 65000, internal link
  BGP version 4, remote router ID 192.168.99.70
  BGP state = Established, up for 00:05:17
  Last read 00:00:09, hold time is 0, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received
    Address family IPv6 Unicast: advertised and received
    ipv6 MPLS Label capability: advertised and received
  Received 54 messages, 0 notifications, 0 in queue
  Sent 55 messages, 1 notifications, 0 in queue
  Default minimum time between advertisement runs is 5 seconds

 For address family: IPv6 Unicast
  BGP table version 21880, neighbor version 21880
  Index 1, Offset 0, Mask 0x2
  Route refresh request: received 0, sent 0
  77 accepted prefixes consume 4928 bytes
  Prefix advertised 4303, suppressed 0, withdrawn 1328
  Number of NLRIs in the update sent: max 1, min 0
```

In the following example, output information linking the MPLS label with prefixes is displayed using the **show mpls forwarding-table** command. If the 6PE feature is configured, the labels are aggregated because there are several prefixes for one local label, and the prefix column contains IPv6 instead of a target prefix.

```
Device# show mpls forwarding-table

Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         Pop Label  1.1.1.1/32       0             Et0/0      10.0.0.1
18         No Label   nh-id(1)         0             Et2/0      10.0.2.2
19         No Label   nh-id(2)         0             Et1/0      10.0.1.2
20         No Label   nh-id(3)         0             Et1/0      10.0.1.2
22         No Label   nh-id(5)         0             Et1/0      10.0.1.2
24         No Label   nh-id(5)         0             Et2/0      10.0.2.2
```

In the following example, output information about the top of the stack label with label switching information is displayed using the **show bgp ipv6** command with the **labels** keyword:

```
Device# show bgp ipv6 labels

Network              Next Hop               In tag/Out tag
2001:DB8:DDDD::/64   ::FFFF:192.168.99.70   notag/20
```

In the following example, output information about labels from the Cisco Express Forwarding table is displayed using the **show ipv6 cef** command with an IPv6 prefix:

```
Device# show ipv6 cef 2001:DB8:DDDD::/64

2001:DB8:DDDD::/64
      nexthop ::FFFF:192.168.99.70
      fast tag rewrite with Se0/0, point2point, tags imposed {19 20}
```

In the following example, output information from the IPv6 routing table is displayed using the **show ipv6 route** command. The output shows the IPv6 MPLS virtual interface as the output interface of IPv6 routes forwarded across the MPLS cloud. This example shows output from the 6PE1 router.

The 6PE2 router has advertised the IPv6 prefix of 2001:DB8:dddd::/48 configured for the CE2 router and the next-hop address is the IPv4-compatible IPv6 address ::ffff:192.168.99.70, where 192.168.99.70 is the IPv4 address of the 6PE2 router.

```
Device# show ipv6 route

IPv6 Routing Table - 10 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
B 2001:DB8:DDDD::/64 [200/0]
  via ::FFFF:192.168.99.70, IPv6-mpls
B 2001:DB8:DDDD::/64 [200/0]
  via ::FFFF:192.168.99.70, IPv6-mpls
L 2001:DB8:FFFF::1/128 [0/0]
  via ::, GigabitEthernet0/0/0
C 2001:DB8:FFFF::/64 [0/0]
  via ::, GigabitEthernet0/0/0
S 2001:DB8:FFFF::/48 [1/0]
  via 2001:DB8:B00:FFFF::2, GigabitEthernet0/0/0
```

Additional References for IPv6 Switching: Provider Edge Router over MPLS

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
IPv6 commands	IPv6 Command Reference
Cisco IOS IPv6 features	IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

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 IPv6 Switching: Provider Edge Router over MPLS

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 9: Feature Information for IPv6 Switching: Provider Edge Router over MPLS

Feature Name	Releases	Feature Information
IPv6 Switching: Provider Edge Router over MPLS	Cisco IOS XE Release 3.1S	The Cisco implementation of IPv6 Provider Edge Router over MPLS enables IPv6 sites to communicate with each other over an MPLS IPv4 core network using MPLS LSPs. The following commands were introduced or modified: **address-family ipv6**, **ipv6 address**, **ipv6 cef**, **ipv6 unicast-routing**, **maximum-paths ibgp**, **neighbor activate**, **neighbor remote-as**, **neighbor send-label**, **neighbor update-source**, **no bgp default ipv4-unicast**, **router bgp**, **show bgp ipv6**, **show bgp ipv6 labels**, **show bgp ipv6 neighbors**, **show ipv6 cef**, **show ipv6 route**, **show mpls forwarding-table**.