MPLS Transport Profile

Last Updated: November 28, 2011

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. 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 feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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-TP

- MPLS-TPPenultimate 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.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

L2VPN Restrictions

- L2VPN interworking is not supported.
- Local switching with AToM pseudowire as a backup is not supported.
- L2VPN pseudowire redundancy to an AToM pseudowire by one or more attachment circuits is not supported.
- PW 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. BFD 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.

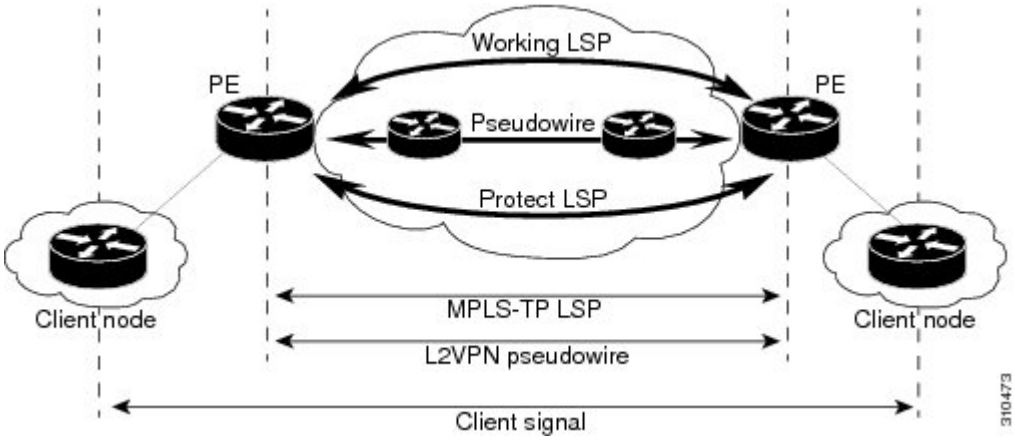
Ping and Trace Restrictions

- 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. An echo reply consists of the following elements:
  - A labeled packet with a GAL label 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.
- 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. See *LSP-Ping Extensions for MPLS-TP* (draft-nitinb-mpls-tp-lsp-ping-extensions-01.txt).

Information About MPLS-TP

How MPLS-TP Works

MPLS-TP tunnels provide the transport network service layer over which IP and MPLS traffic traverse. MPLS-TP tunnels help transition from SONET/SDH 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 LSPs support 1-to-1 path protection. You can configure the working and protect LSPs as part of configuring the MPLS-TP tunnel. The working LSP is the primary LSP used to route traffic. The protect LSP is 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

MPLS-TP LSPs are bidirectional and co-routed and are comprised of two unidirectional LSPs that are supported by the MPLS forwarding infrastructure. A TP tunnel consists of a pair of unidirectional tunnels providing a bidirectional LSP. Each unidirectional tunnel can optionally be protected with a protect LSP that activates automatically upon failure conditions.

MPLS-TP OAM Support

Several OAM protocols and messages support the provisioning and maintenance of MPLS-TP tunnels and bidirectional LSPs:

- MPLS-TP OAM: GACH: Generic Associated Channel (G-ACh) is the control channel mechanism associated with 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 is used to support in-band OAMs of MPLS LSPs and PWs. The OAM messages are used for fault management, connection verification, continuity check and other functions.

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

- ◦ OAM Fault Management: AIS, LDI and LKR messages. (GAL with fault-OAM channel)
  ◦ OAM Connection Verification: ping and traceroute messages. (GAL with IP channel by default)
  ◦ OAM Continuity Check: BFD (non-IP BFD and IP BFD) messages. (GAL with BFD channel or IP channel depending on message format)

The following messages are forwarded along the specified PW:

- ◦ Static PW OAM messages (static PW status)
  ◦ PW ping and traceroute messages
  ◦ PW BFD messages
- MPLS-TP OAM: Fault Management: Link Down Indication (LDI), Alarm Indication Signal (AIS), and Lock Report (LKR) messages. LDI messages are generated at midpoint nodes when a failure is detected. At the midpoint, an LDI message will be sent to the endpoint that is reachable with the existing failure. Similarly, LKR messages will be sent from a midpoint node to the reachable endpoint when an interface is administratively shut. AIS messages are not generated by Cisco, but are processed if received. By default, reception of LDI and LKR on the active LSP at an endpoint will cause a path protection switchover, while AIS will not.
- MPLS-TP OAM: Fault Management: Emulated Protection Switching for LSP Lockout. Cisco implements a form of Emulated Protection Switching in support of LSP Lockout using customized Fault messages. When a Cisco 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 up so that BFD and other OAM messages can continue to traverse it. Maintenance of the LSP can take place (such as reconfiguring or replacing a midpoint LSR). The LSP is shown as UP and OAM can verify connectivity before the LSP is put back into service by removing the lockout. Lockout of the working LSP is not allowed if no protect LSP is configured. Alternatively, lockout of the protect LSP is allowed if no working LSP is configured.
- LSP ping and trace: For MPLS-TP connectivity verification, you can use **ping mpls tp** and **trace mpls tp**commands. You can specify that the echo requests be sent along either the working LSP, the protect LSP, or the active LSP. You can also specify that the echo request be sent on a locked out MPLS-TP tunnel LSP (either working or protect) if the working or protect LSP is explicitly specified.
- MPLS-TP OAM: Continuity Check via BFD: You can configure BFD sessions running over MPLS-TP LSPs. BFD sessions run on both the working LSP and the protect LSP. In order to perform a path protection switchover within 60 msec on an MPLS-TP endpoint, the BFD Hardware Offload feature enables the router hardware to construct and send BFD messages, which removes the task from the software path. You do not need to configure the BFD Hardware Offload feature. It works automatically on supported platforms. You must enable BFD.

MPLS-TP Static and Dynamic Multisegment Pseudowires

MPLS-TP supports the following combinations of static and dynamic multisegment pseudowires:

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

MPLS-TP L2VPN Pseudowire Redundancy for Static and Dynamic Multisegment Pseudowires

MPLS-TP supports one-to-one L2VPN pseudowire redundancy for the following combinations of static and dynamic pseudowires:

- Static pseudowire with a static backup pseudowire
- Static pseudowire with a dynamic backup pseudowire
- Dynamic pseudowire with a static backup pseudowire

MPLS-TP OAM Status for Static and Dynamic Multisegment Pseudowires

With static pseudowires, status notifications can be provided by BFD over VCCV or 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, static pseudowire OAM protocol is preferred. You can acquire per pseudowire OAM for attachment circuit/pseudowire notification over VCCV channel with or without the control word.

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.

The MPLS-TP link is used to create a level 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 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 Configuring MPLS-TP Links and Physical Interfaces, page 19 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 the information for the LSP is specified with the **mpls tp lsp** command, which enters the submode 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 router and a coworker's router, then your router is the source. However, your coworker considers his or her router to be the source. At the midpoint, either router 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 router ID and global ID, or from locally configured information using the **tp source**command after you enter the command **interface tunnel-tp** *number*command, where *number* is the local/source tunnel-number.

- At the endpoint, the remote information (destination) is configured using the **tp destination** command after you enter the command **interface tunnel-tp** *number*. The **tp destination** command includes the destination node ID, optionally the global ID, and optionally 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 the LSPs at the midpoint routers, make that the configuration does not reflect traffic back to the originating node.

How to Configure MPLS-TP

Configuring the MPLS Label Range

You must specify a static range of 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*}

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label range** *minimum-value maximum-value* {**static** *minimum-static-value maximum-static-value*} **Example:** `Router(config)# mpls label range 1001 1003 static 10000 25000`	Specifies a static range of MPLS labels

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*

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 tp** **Example:** `Router(config)# mpls tp`	Enters MPLS-TP configuration mode, from which you can configure MPLS-TP parameters for the router.

	Command or Action	Purpose
Step 4	**router-id** *node-id* **Example:** `Router(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 router.
Step 5	**global-id** *num* **Example:** `Router(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 by routers at tunnel midpoints to help isolate the location of faults.

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 can provide 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*]

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.

	Command or Action	Purpose
Step 3	**bfd-template single-hop** *template-name* **Example:** `Router(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:** `Router(config-bfd)# interval min-tx 99 min-rx 99 multiplier 3`	Specifies a set of BFD interval values.

Configuring Pseudowire OAM Attributes

SUMMARY STEPS

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

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	**pseudowire-static-oam class** *class-name* **Example:** `Router(config)# pseudowire-static-oam class oam-class1`	Creates a pseudowire OAM class and enters pseudowire OAM class configuration mode.

	Command or Action	Purpose
Step 4	**timeout refresh send** *seconds* **Example:** `Router(config-st-pw-oam-class)# timeout refresh send 20`	Specifies the OAM timeout refresh intervals.

Configuring the Pseudowire Class

When you create the 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**]

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	**pseudowire-class** *class-name* **Example:** `Router(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:** `Router(config-pw-class)# encapsulation mpls`	Specifies the encapsulation type.
Step 5	**control-word** **Example:** `Router(config-pw-class)# control-word`	Enables the use of the control word.
Step 6	**protocol** {**l2tpv2** | **l2tpv3** | **none**} [*l2tp-class-name*] **Example:** `Router(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:** `Router(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:** `Router(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:** `Router(config-pw-class)# vccv bfd template bfd-temp1 raw-bfd`	Specifies the VCCV BFD template to use.

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*]

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	**interface** *type number* **Example:** `Router(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:** `Router(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:** `Router(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:** `Router(config-if-xconn)# no mpls control-word`	Specifies the control word.
Step 7	**backup delay** {*enable-delay-period* | **never**} {*disable-delay-period* | **never**} **Example:** `Router(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:** `Router(config-if-xconn)# backup peer 10.0.0.2 50`	Specifies a redundant peer for a pseudowire virtual circuit (VC).

Configuring the MPLS-TP Tunnel

On the endpoint routers, 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. **exit**

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	**interface tunnel-tp** *number* **Example:** `Router(config)# interface tunnel-tp 2`	Enters tunnel interface configuration mode. Tunnel numbers from 0 to 999 are supported.

	Command or Action	Purpose
Step 4	**description** *tunnel-description* **Example:** `Router(config-if)# description headend tunnel`	(Optional) Specifies a tunnel description.
Step 5	**tp tunnel-name** *name* **Example:** `Router(config-if)# tp tunnel-name tunnel22`	Specifies the name of the MPLS-TP tunnel. The TP tunnel name is displayed in the **show mpls tp tunnel**command output. This command is useful for consistently identifying the tunnel at all endpoints and midpoints.
Step 6	**tp bandwidth** *num* **Example:** `Router(config-if)# tp bandwidth 10000`	Specifies the tunnel bandwidth.
Step 7	**tp source** *node-id* [*global-id num*] **Example:** `Router(config-if)# tp source 10.10.11.11 global-id 10`	(Optional) Specifies the tunnel source and endpoint. This command is and not typically used, because the global router ID and global ID can be used to identify the tunnel source at the endpoint. All tunnels on the router generally use the same (globally specified) source information.
Step 8	**tp destination** *node-id* [[**tunnel-tp** *num*] **global-id** *num*] **Example:** `Router(config-if)# tp destination 10.10.10.10`	Specifies the destination node of the tunnel.
Step 9	**bfd** *bfd-template* **Example:** `Router(config-if)# bfd mpls-tp-bfd-2`	Specifies the BFD template.
Step 10	**working-lsp** **Example:** `Router(config-if)# working-lsp`	Specifies a working LSP, also known as the primary LSP. This LSP is used to route traffic. This command enters working LSP interface configuration mode (config-if-working).

	Command or Action	Purpose
Step 11	**in-label** *num* **Example:** `Router(config-if-working)# in-label 111`	Specifies the in label.
Step 12	**out-label** *num* **out-link** *num* **Example:** `Router(config-if-working)# out-label 112 out-link 1`	Specifies the out label and out link.
Step 13	**exit** **Example:** `Router(config-if-working)# exit`	Exits from working LSP interface configuration mode.
Step 14	**protect-lsp** **Example:** `Router(config-if)# protect-lsp`	Specifies 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. This command enters protect LSP interface configuration mode (config-if-protect).
Step 15	**in-label** *num* **Example:** `Router(config-if-protect)# in-label 100`	Specifies the in label.
Step 16	**out-label** *num* **out-link** *num* **Example:** `Router(config-if-protect)# out-label 113 out-link 2`	Specifies the out label and out link.
Step 17	**exit** **Example:** `Router(config-if-protect)# exit`	Exits from protect LSP interface configuration mode.

Configuring MPLS-TP LSPs at Midpoints

Note When configuring the LSPs at the midpoint routers, make that the configuration does not reflect 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*

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 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:** `Router(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.

	Command or Action	Purpose
Step 4	**forward-lsp** **Example:** `Router(config-mpls-tp-lsp)# forward-lsp`	Enters MPLS-TP LSP forward LSP configuration mode.
Step 5	bandwidth *num* **Example:** `Router(config-mpls-tp-lsp-forw)# bandwidth 100`	Specifies the bandwidth.
Step 6	**in-label** *num* **out-label** *num* **out-link** *num* **Example:** `Router(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:** `Router(config-mpls-tp-lsp-forw)# exit`	Exits MPLS-TP LSP forward LSP configuration mode.
Step 8	**reverse-lsp** **Example:** `Router(config-mpls-tp-lsp)# reverse-lsp`	Enters MPLS-TP LSP reverse LSP configuration mode.
Step 9	bandwidth *num* **Example:** `Router(config-mpls-tp-lsp-rev)# bandwidth 100`	Specifies the bandwidth.
Step 10	**in-label** *num* **out-label** *num* **out-link** *num* **Example:** `Router(config-mpls-tp-lsp-rev)# in-label 33 out-label 23 out-link 44`	Specifies the in label, out label, and out link numbers.

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/num*
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. **exit**
8. **exit**
9. **show mpls tp link-numbers**

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	**interface** *type/num* **Example:** `Router(config)# interface ethernet 1/0`	Specifies the interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip address** *ip-address mask* **Example:** `Router(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:** `Router(config-if)# mpls tp link 1 ipv4 10.0.0.2`	Associates an MPLS-TP link number with a physical interface and next-hop node. On point-to-point interfaces or Ethernet interfaces designated as point-to-point using the **medium p2p** command, the next-hop can be implicit, so the **mpls tp link**command just associates a link number to the interface. Multiple tunnels and LSPs can refer to the MPLS-TP link to indicate they are traversing that interface. You can move the MPLS-TP link from one interface to another without reconfiguring all the MPLS-TP tunnels and LSPs that refer to the link. Link numbers a must be unique on the router 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:** `Router(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	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 8	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

	Command or Action	Purpose
Step 9	**show mpls tp link-numbers** **Example:** `Router# 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**

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	**l2 vfi** *name* **point-to-point** **Example:** `Router(config)# l2 vfi atom point-to-point`	Creates a point-to-point Layer 2 virtual forwarding interface (VFI) and enters VFI configuration mode.

	Command or Action	Purpose
Step 4	**neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} **Example:** `Router(config-vfi)# neighbor 10.111.111.111 123 pw-class atom`	Sets up an emulated VC. Specify the IP address and the VC ID of the remote router. Also specify the pseudowire class to use for the emulated VC. Note: Only two **neighbor** commands are allowed for each l2 vfi point-to-point command.
Step 5	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Router(config-vfi)# mpls label 101 201`	Configures the static pseudowire connection by defining local and remote circuit labels.
Step 6	**mpls control-word** **Example:** `Router(config-vfi)# mpls control-word`	Specifies the control word.
Step 7	**neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} **Example:** `Router(config-vfi)#`	Sets up an emulated VC. Specify the IP address and the VC ID of the remote router. Also specify the pseudowire class to use for the emulated VC.
Step 8	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Router(config-vfi)# Router(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:** `Router(config-vfi)# mpls control-word`	Specifies the control word.

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*

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	**pseudowire-tlv template** *template-name* **Example:** `Router(config)# pseudowire-tlv template statictemp`	Creates a template of pseudowire type-length-value (TLV) parameters
Step 4	**tlv** [*type-name*] *type-value length* [**dec** | **hexstr** | **str**] *value* **Example:** `Router(config-pw-tlv-template)# tlv statictemp 2 4  hexstr 1`	Specifies the TLV parameters.

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, 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*

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	**pseudowire-class** *class-name* **Example:** `Router(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:** `Router(config-pw-class)# encapsulation mpls`	Specifies the encapsulation type.
Step 5	**control-word** **Example:** `Router(config-pw-class)# control-word`	Enables the use of the control word.
Step 6	**protocol** {**l2tpv2** | **l2tpv3** | **none**} [*l2tp-class-name*] **Example:** `Router(config-pw-class)# protocol none`	Specifies the type of protocol. Use the **protocol none** command to specify a static pseudowire.
Step 7	**exit** **Example:** `Router(config-pw-class)# exit`	Exits pseudowire class configuration mode.
Step 8	**pseudowire-class** *class-name* **Example:** `Router(config)# pseudowire-class mpls-tp-class1`	Creates a pseudowire class and enters pseudowire class configuration mode.
Step 9	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies the encapsulation type.
Step 10	**exit** **Example:** `Router(config-pw-class)# exit`	Exits pseudowire class configuration mode.

Command or Action	Purpose
Step 11 **l2 vfi** *name* **point-to-point** **Example:** `Router(config)# l2 vfi atom point-to-point`	Creates a point-to-point Layer 2 virtual forwarding interface (VFI) and enters VFI configuration mode.
Step 12 **neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} **Example:** `Router(config-vfi)# neighbor 10.111.111.111 123 pw-class atom`	Sets up an emulated VC. Specify the IP address and the VC ID of the remote router. Also specify the pseudowire class to use for the emulated VC. Enters config-vfi-neighbor command mode. 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:** `Router(config-vfi-neighbor)# neighbor 10.111.111.111 123 pw-class atom`	Sets up an emulated VC. Specify the IP address and the VC ID of the remote router. Also specify the pseudowire class to use for the 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:** `Router(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:** `Router(config-vfi-neighbor)# mpls control-word`	Specifies the control word.
Step 16 **local interface** *pseudowire-type* **Example:** `Router(config-vfi-neighbor)# local interface 4`	Specifies the pseudowire type and enters VFI neighbor interface configuration mode.

Command or Action	Purpose
Step 17 Do one of the following: • **tlv** [*type-name*] *type-value length* [**dec** | **hexstr** | **str**] *value* • • **tlv template** *template-name* **Example:** `Router(config-vfi-neighbor)# tlv statictemp 2 4 hexstr 1`	Specifies the TLV parameters or invokes a previously configured TLV template.

• Example, page 27

Example

```
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
```

Configuring the L2VPN Pseudowire Redundancy for Static Multisegment Pseudowires

Perform the following steps to configure the L2VPN pseudowire redundancy for static multisegment pseudowires that are backed up with static or dynamic multisegment pseudowires.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface ethernet** *type/num*
4. **service instance** *id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **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**}]
7. **mpls label** *local-pseudowire-label remote-pseudowire-label*
8. **mpls control-word**
9. **backup delay** {*enable-delay-period* | **never**} {*disable-delay-period* | **never**}
10. **backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*] [**priority** *value*]
11. **mpls label** *local-pseudowire-label remote-pseudowire-label*
12. **mpls control-word**

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	**interface ethernet** *type/num* **Example:** `Router(config)# interface ethernet 1/0`	Specifies the interfaces and enters interface configuration mode.
Step 4	**service instance** *id* **ethernet** **Example:** `Router(config-if)# service instance 1 ethernet`	Specifies the service instance and enters service instance interface configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 10`	Enables the interface to accept 802.1Q VLAN packets.
Step 6	**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:** `Router(config-if-srv)# xconnect 10.109.10.10 123encapsulation mpls manual pw-class stpw`	Binds the attachment circuit to a pseudowire VC and enters xconnect configuration mode.
Step 7	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Router(cfg-if-ether-vc-xconn)# mpls label 100 150`	Configures the static pseudowire connection by defining local and remote circuit labels.

	Command or Action	Purpose
Step 8	**mpls control-word** **Example:** `Router(cfg-if-ether-vc-xconn)# no mpls control-word`	Specifies the control word.
Step 9	**backup delay** {*enable-delay-period* | **never**} {*disable-delay-period* | **never**} **Example:** `Router(cfg-if-ether-vc-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 10	**backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*] [**priority** *value*] **Example:** `Router(cfg-if-ether-vc-xconn)# backup peer 10.0.0.2 50`	Specifies a redundant peer for a pseudowire virtual circuit (VC). Enters backup xconnect configuration mode.
Step 11	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Router(cfg-if-ether-vc-xconn-bkup)# mpls label 100 150`	Configures the static pseudowire connection by defining local and remote circuit labels.
Step 12	**mpls control-word** **Example:** `Router(cfg-if-ether-vc-xconn-bkup)# no mpls control-word`	Specifies the control word.

- Example, page 29

Example

```
interface Ethernet1/0
 no ip address
 no shutdown
 service instance 1 ethernet
  encapsulation dot1q 10
  xconnect 10.113.113.113 123 encapsulation mpls manual pw-class stpw
   mpls label 0 101
   mpls control-word
   backup peer 1 0.120.120.120 124 pw-class stpw
    mpls label 0 105
    mpls control-word
```

Verifying the MPLS-TP Configuration

When the entire tunnel is programmed, use the following commands to verify and help troubleshoot the configuration:

- **show mpls tp tunnel-tp lsps:** To ensure that both LSPs are up and working from a tunnel endpoint.
- **show mpls tp tunnel-tp** *number* **detail** : To help determine the cause if the tunnel is not up and working.
- **show bfd neighbors mpls-tp:** To display the state of BFD, which must be up for the endpoint LSPs to be up.
- **trace mpls tp** and **ping mpls tp**: To help isolate any connectivity issues.
- **debug mpls tp** : To enable the display of MPLS-TP error messages.
- **logging** (MPLS-TP): To enable the display of logging messages related to configuration changes or state changes.
- **show mpls l2transport static-oam: To enable the display of MPLS-TP messages related to pseudowires.**

Additional References

Related Documents

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

Standards

Standard	Title
draft-ietf-mpls-tp-gach-gal-xx	*MPLS Generic Associated Channel*

MIBs

MIB	MIBs Link
None	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
RFC 5921	*A Framework for MPLS in Transport Networks*
RFC 5885	*Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)*
RFC 5586	*MPLS Generic Associated Channel*

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-TP

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-TP

Feature Name	Releases	Feature Information
MPLS Transport Profile	15.1(1)SA 15.3(1)S 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 traverse. 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. 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 mpls tp pseudowire-static-oam class pseudowire-tlv template show mpls l2transport static-oam show mpls tp status protocol notification static timeout refresh tlv tlv template trace mpls tp**
L2VPN Static to Dynamic PW Interconnection & PW Preferred Path for MPLS-TP Tunnels	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.
MPLS-TP: PW Redundancy for Static PWs	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.
Bidirectional MPLS-TP LSP	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.

Feature Name	Releases	Feature Information
MPLS-TP Path Protection	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.
MPLS-TP OAM: GACH	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.
MPLS-TP OAM: Continuity Check via BFD	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.
MPLS-TP OAM: Ping/Trace	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.
MPLS-TP OAM: Fault Management	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.
MPLS TP: IP-less configuration of MPLS TP tunnels	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.