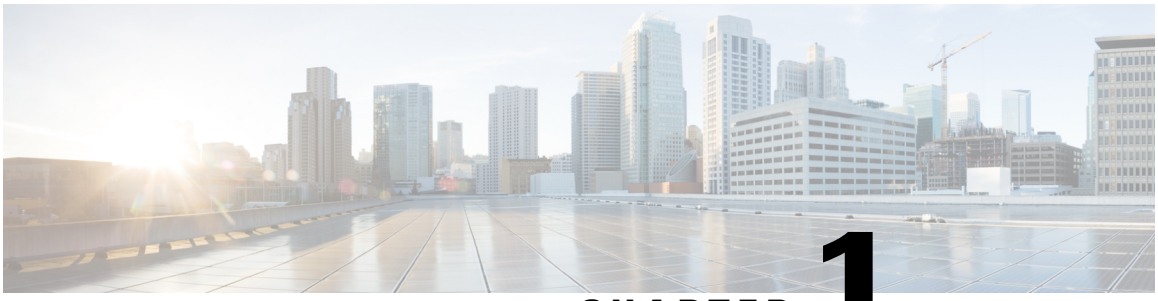
MPLS Layer 2 VPNs Configuration Guide, Cisco IOS Release 15SY

First Published: November 26, 2012

Last Modified: November 26, 2012

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

Any Transport over MPLS

This document describes the Any Transport over MPLS (AToM) feature, which provides the following capabilities:

- Transport data link layer (Layer2) packets over a Multiprotocol Label Switching (MPLS) backbone.
- Enable service providers to connect customer sites with existing Layer 2 networks by using a single, integrated, packet-based network infrastructure--a Cisco MPLS network. Instead of using separate networks with network management environments, service providers can deliver Layer 2 connections over an MPLS backbone.
- Provide a common framework to encapsulate and transport supported Layer 2 traffic types over an MPLS network core.

AToM supports the following like-to-like transport types:

- ATM Adaptation Layer Type-5 (AAL5) over MPLS
- ATM Cell Relay over MPLS
- Ethernet over MPLS (VLAN and port modes)
- Frame Relay over MPLS
- PPP over MPLS
- High-Level Data Link Control (HDLC) 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 at the end of this module.

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

Prerequisites for Any Transport over MPLS

Before configuring AToM, ensure that the network is configured as follows:

- Configure IP routing in the core so that the provider edge (PE) routers can reach each other via IP.
- Configure MPLS in the core so that a label-switched path (LSP) exists between the PE routers.
- Enable Cisco Express Forwarding or distributed Cisco Express Forwarding before configuring any Layer 2 circuits.
- Configure a loopback interface for originating and terminating Layer 2 traffic. Make sure the PE routers can access the other router's loopback interface. Note that the loopback interface is not needed in all cases. For example, tunnel selection does not need a loopback interface when AToM is directly mapped to a traffic engineering (TE) tunnel.
- AToM is supported on the Cisco 7200 and 7500 series routers. For details on supported hardware, see the following documents:
  - Cross-Platform Release Notes for Cisco IOS Release 12.0S
  - Cross-Platform Release Notes for Cisco IOS Release 12.4T, Part 2: Platform-Specific Information
- AToM is supported on the Cisco 7600 routers. For details on supported shared port adapters and line cards, see the following documents:
  - Guide to Supported Hardware for Cisco 7600 Series Routers with Release 12.2SR
  - Cross-Platform Release Notes for Cisco IOS Release 12.2SR for the Cisco 7600 Series Routers
- The Cisco 7600 router has platform-specific instructions for configuring some AToM features. Platform-specific configuration information is included in the following documents:
  - The "Configuring PFC3BXL and PFC3B Mode Multiprotocol Label Switching" module of the Cisco 7600 Series Cisco IOS Software Configuration Guide, Release 12.2SR
  - The "Configuring Multiprotocol Label Switching on the Optical Services Modules" module of the OSM Configuration Note , Release 12.2SR
  - The "Configuring Multiprotocol Label Switching on FlexWAN and Enhanced FlexWAN Modules" module of the FlexWAN and Enhanced FlexWAN Modules Installation and Configuration Guides of Cisco 7600 Series Routers

  - The “Configuring Any Transport over MPLS on a SIP” section of the Cisco 7600 Series Router SIP, SSC, and SPA Software Configuration Guide
  - The “Configuring AToM VP Cell Mode Relay Support” section of the Cisco 7600 Series Router SIP, SSC, and SPA Software Configuration Guide
  - The Cross-Platform Release Notes for Cisco IOS Release 12.2SR
- AToM is supported on the Cisco 10000 series routers. For details on supported hardware, see the “Configuring Any Transport over MPLS” section of the Cisco 10000 Series Router Software Configuration Guide.
- The Cisco 10000 series router has platform-specific instructions for configuring some AToM features. Platform-specific configuration information is contained in the “Configuring Any Transport over MPLS” section of the Cisco 10000 Series Router Broadband Aggregation, Leased-Line, and MPLS Configuration Guide.
- AToM is supported on the Cisco12000 series routers. For information about hardware requirements, see the Cross-Platform Release Notes for Cisco IOS Release 12.0S.

Restrictions for Any Transport over MPLS

General Restrictions

The following general restrictions pertain to all transport types under AToM:

- Address format: Configure the Label Distribution Protocol (LDP) router ID on all PE routers to be a loopback address with a /32 mask. Otherwise, some configurations might not function properly.

Ethernet over MPLS (EoMPLS) Restrictions

The following restrictions pertain to the Ethernet over MPLS feature:

- Ethernet over MPLS supports VLAN packets that conform to the IEEE 802.1Q standard. The 802.1Q specification establishes a standard method for inserting VLAN membership information into Ethernet frames. The Inter-Switch Link (ISL) protocol is not supported between the PE and CE routers.
- The AToM control word is supported. However, if the peer PE does not support a control word, the control word is disabled. This negotiation is done by LDP label binding.
- Ethernet packets with hardware-level cyclic redundancy check (CRC) errors, framing errors, and runt packets are discarded on input.

Information About Any Transport over MPLS

How AToM Transports Layer 2 Packets

AToM encapsulates Layer 2 frames at the ingress PE and sends them to a corresponding PE at the other end of a pseudowire, which is a connection between the two PE routers. The egress PE removes the encapsulation and sends out the Layer 2 frame.

The successful transmission of the Layer 2 frames between PE routers is due to the configuration of the PE routers. You can set up the connection, called a pseudowire, between the routers and specify the following information on each PE router:

- The type of Layer 2 data that will be transported across the pseudowire such as Ethernet, Frame Relay, or ATM
- The IP address of the loopback interface of the peer PE router, which enables the PE routers to communicate
- A unique combination of peer PE IP address and VC ID that identifies the pseudowire

The following example shows the basic configuration steps on a PE router that enable the transport of Layer 2 packets. Each transport type has slightly different steps.

Step 1 defines the interface or subinterface on the PE router:

```
Router# interface
interface-type interface-number
```

Step 2 specifies the encapsulation type for the interface, such as dot1q:

```
Router(config-if)# encapsulation encapsulation-type
```

Step 3 does the following:

- Makes a connection to the peer PE router by specifying the LDP router ID of the peer PE router.
- Specifies a 32-bit unique identifier, called the VC ID, which is shared between the two PE routers.

The combination of the peer router ID and the VC ID must be unique on the router. Two circuits cannot use the same combination of the peer router ID and VC ID.

- Specifies the tunneling method used to encapsulate data in the pseudowire. AToM uses MPLS as the tunneling method.

```
Router(config-if)# xconnect peer-router-id vcid encapsulation mpls
```

As an alternative, you can set up a pseudowire class to specify the tunneling method and other characteristics. For more information, see the Configuring the Pseudowire Class, on page 14.

AToM Configuration Commands Prior to Cisco IOS Release 12.0(25)S

In releases of AToM before Cisco IOS 12.0(25)S, the **mpls l2 transport route**command was used to configure AToM circuits.This command has been replaced with the **xconnect**command.

No enhancements will be made to the **mpls l2transport route**command. Enhancements will be made to either the **xconnect**command or the **pseudowire-class**command. Therefore, Cisco recommends that you use the **xconnect** command to configure AToM circuits.

Configurations from releases before Cisco IOS 12.0(25)S that use the **mpls l2transport route**command are still supported.

Benefits of AToM

The following list explains some of the benefits of enabling Layer 2 packets to be sent in the MPLS network:

- The AToM product set accommodates many types of Layer 2 packets, including Ethernet and Frame Relay, across multiple Cisco router platforms, such as the Cisco 7200 and Cisco 7500 series routers. This enables the service provider to transport all types of traffic over the backbone and accommodate all types of customers.
- AToM adheres to the standards developed for transporting Layer 2 packets over MPLS. (See the "Standards" section for the specific standards that AToM follows.) This benefits the service provider that wants to incorporate industry-standard methodologies in the network. Other Layer 2 solutions are proprietary, which can limit the service provider's ability to expand the network and can force the service provider to use only one vendor's equipment.
- Upgrading to AToM is transparent to the customer. Because the service provider network is separate from the customer network, the service provider can upgrade to AToM without disruption of service to the customer. The customers assume that they are using a traditional Layer 2 backbone.

MPLS Traffic Engineering Fast Reroute

AToM can use MPLS traffic engineering (TE) tunnels with fast reroute (FRR) support. AToM VCs can be rerouted around a failed link or node at the same time as MPLS and IP prefixes.

Enabling fast reroute on AToM does not require any special commands; you can use the standard fast reroute (FRR) commands. At the ingress PE, an AToM tunnel is protected by fast reroute when it is routed to an FRR-protected TE tunnel. Both link and node protection are supported for AToM VCs at the ingress PE. For more information on configuring MPLS TE fast reroute, see the following document:

MPLS Traffic Engineering (TE)--Link and Node Protection, with RSVP Hellos Support

Note The AToM VC independence feature was introduced in Cisco IOS Release 12.0(31)S. This feature enables the Cisco 12000 series router to perform fast reroute in fewer than 50 milliseconds, regardless of the number of VCs configured. In previous releases, the fast reroute time depended on the number of VCs inside the protected TE tunnel.

For the Cisco 12000 series routers, fast reroute uses three or more labels, depending on where the TE tunnel ends:

- If the TE tunnel is from a PE router to a PE router, three labels are used.
- If the TE tunnel is from a PE router to the core router, four labels are used.

Engine 0 ATM line cards support three or more labels, but the performance degrades. Engine 2 Gigabit Ethernet line cards and engine 3 line cards support three or more labels and can work with the fast reroute feature.

You can issue the **debug mpls l2transport fast-reroute**command to debug fast reroute with AToM.

Note This command does not display output on platforms where AToM fast reroute is implemented in the forwarding code. The command does display output on Cisco 10720 Internet router line cards and Cisco 12000 series line cards. This command does not display output for the Cisco 7500 (both Route Processor (RP) and Versatile Interface Processor (VIP)) series routers, Cisco 7200 series routers, and Cisco 12000 series RP.

In the following example, the primary link is disabled, which causes the backup tunnel (Tunnel 1) to become the primary path. In the following example, bolded output shows the status of the tunnel:

```
Router# execute-on slot 3 debug mpls l2transport fast-reroute
========= Line Card (Slot 3) =========
AToM fast reroute debugging is on
SLOT 3:Sep 16 17:58:56.346: AToM SMGR: Processing TFIB FRR event for 10.4.0.1
SLOT 3:Sep 16 17:58:56.346: AToM SMGR: Finished processing TFIB FRR event for 10.4.0.1
SLOT 3:Sep 16 17:58:56.346: AToM SMGR: Processing TFIB FRR event for Tunnel41
SLOT 3:Sep 16 17:58:56.346: AToM SMGR: Finished processing TFIB FRR event for Tunnel41
Sep 16 17:58:58.342: %LINK-3-UPDOWN: Interface POS0/0, changed state to down
Sep 16 17:58:58.342: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.1 on POS0/0 from FULL to DOWN,
Neighbor Down: Interface down or detached
Sep 16 17:58:59.342: %LINEPROTO-5-UPDOWN: Line protocol on Interface POS0/0, changed state
 to down
```

Maximum Transmission Unit Guidelines for Estimating Packet Size

The following calculation helps you determine the size of the packets traveling through the core network. You set the maximum transmission unit (MTU) on the core-facing interfaces of the P and PE routers to accommodate packets of this size. The MTU should be greater than or equal to the total bytes of the items in the following equation:

Core MTU >= (Edge MTU + Transport header + AToM header + (MPLS label stack * MPLS label size))

The following sections describe the variables used in the equation:

Edge MTU

The edge MTU is the MTU for customer-facing interfaces.

Transport Header

The Transport header depends on the transport type. The table below lists the specific sizes of the headers.

Table 1: Header Size of Packets

Transport Type	Packet Size
AAL5	0-32 bytes
Ethernet VLAN	18 bytes

Transport Type	Packet Size
Ethernet Port	14 bytes
Frame Relay DLCI	2 bytes for Cisco encapsulation, 8 bytes for Internet Engineering Task Force (IETF) encapsulation
HDLC	4 bytes
PPP	4 bytes

AToM Header

The AToM header is 4 bytes (control word). The control word is optional for Ethernet, PPP, HDLC, and cell relay transport types. However, the control word is required for Frame Relay and ATM AAL5 transport types.

MPLS Label Stack

The MPLS label stack size depends on the configuration of the core MPLS network:

- AToM uses one MPLS label to identify the AToM VCs (VC label). Therefore, the minimum MPLS label stack is one for directly connected AToM PEs, which are PE routers that do not have a P router between them.
- If LDP is used in the MPLS network, the label stack size is two (the LDP label and the VC label).
- If a TE tunnel is used instead of LDP between PE routers in the MPLS network, the label stack size is two (the TE label and the VC label).
- If a TE tunnel and LDP are used in the MPLS network (for example, a TE tunnel between P routers or between P and PE routers, with LDP on the tunnel), the label stack is three (the TE label, LDP label, and VC label).
- If you use MPLS fast reroute in the MPLS network, you add a label to the stack. The maximum MPLS label stack in this case is four (the FRR label, TE label, LDP label, and VC label).
- If AToM is used by the customer carrier in an MPLS VPN Carrier Supporting Carrier environment, you add a label to the stack. The maximum MPLS label stack in the provider carrier network is five (the FRR label, TE label, LDP label, VPN label, and VC label).
- If an AToM tunnel spans different service providers that exchange MPLS labels using IPv4 Border Gateway Protocol (BGP) (RFC 3107), you add a label to the stack. The maximum MPLS label stack is five (the FRR label, TE label, Border Gateway Protocol (BGP) label, LDP label, and VC label).

Other circumstances can increase the MPLS label stack size. Therefore, analyze the complete data path between the AToM tunnel endpoints, determine the maximum MPLS label stack size for your network, and then multiply the label stack size by the size of the MPLS label.

Example Estimating Packet Size

The size of packets is estimated in the following example, which uses the following assumptions:

- The edge MTU is 1500 bytes.

- The transport type is Ethernet VLAN, which designates 18 bytes for the transport header.
- The AToM header is 0, because the control word is not used.
- The MPLS label stack is 2, because LDP is used. The MPLS label is 4 bytes.

```
Edge MTU + Transport header + AToM header + (MPLS label stack * MPLS label) = Core MTU
1500     + 18                 + 0          + (2                * 4          ) = 1526
```

You must configure the P and PE routers in the core to accept packets of 1526 bytes.

Once you determine the MTU size to set on your P and PE routers, you can issue the **mtu**command on the routers to set the MTU size. The following example specifies an MTU of 1526 bytes:

```
Router(config-if)# mtu 1526
```

mpls mtu Command Changes

Some interfaces (such as FastEthernet) require the **mpls mtu** command to change the MTU size. In Cisco IOS Release 12.2(25)S, the behavior of the **mpls mtu** command changed.

If the interface MTU is fewer than 1524 bytes, you can set the maximum MPLS MTU to 24 bytes more than the interface MTU. For example, if the interface MTU is set to 1510 bytes, then you can set the maximum MPLS MTU to 1534 bytes (1510 + 24).

Caution Although you can set the MPLS MTU to a value greater than the interface MTU, you must set the MPLS MTU value to less than or equal to the interface MTU to prevent data corruption, dropped packets, and high CPU rates.

If the interface MTU is greater than or equal to 1524 bytes, then you can set the maximum MPLS MTU value to as high as the interface MTU value. For example, if the interface MTU is set to 1600 bytes, then you can set the MPLS MTU to a maximum of 1600 bytes. If you set the MPLS MTU value to higher than the interface MTU, traffic is dropped.

For interfaces that do not allow you to configure the interface MTU value and for interfaces where the interface MTU is 1500 bytes, the MPLS MTU range is 64 to 1524 bytes.

For GRE tunnel interfaces you can set the MPLS MTU value to either the default value or the maximum value that is supported by the platform for the interface.

You can set the MPLS MTU value to the maximum value by using the **max** keyword along with the **mpls mtu** command. The **mpls mtu max** command allows the previously dropped packets to pass through the GRE tunnel by fragmentation on the underlying physical interface.

Note that the MPLS MTU value cannot be greater than the interface MTU value for non-GRE tunnels.

If you upgrade to Cisco IOS Release 12.2(25)S and you have an MPLS MTU setting that does not conform to these guidelines, the command is rejected.

For Cisco IOS Release 12.2(27)SBC, 12.2(33)SRA, 12.4(11)T, 12.2(33)SXH, and later releases, you cannot set the MPLS MTU to a value greater than the interface MTU. This eliminates problems, such as dropped packets, data corruption, and high CPU rates. See the MPLS MTU Command Changes document for more information.

Per-Subinterface MTU for Ethernet over MPLS

MTU values can be specified in xconnect subinterface configuration mode. When you use xconnect subinterface configuration mode to set the MTU value, you establish a pseudowire connection for situations where the interfaces have different MTU values that cannot be changed.

If you specify an MTU value in xconnect subinterface configuration mode that is outside the range of supported MTU values (64 bytes to the maximum number of bytes supported by the interface), the command might be rejected. If you specify an MTU value that is out of range in xconnect subinterface configuration mode, the router enters the command in subinterface configuration mode.

For example, if you specify an MTU of 1501 in xconnect subinterface configuration mode, and that value is out of range, the router enters the command in subinterface configuration mode, where it is accepted:

```
Router# configure terminal
Router(config)# interface gigabitethernet0/0/2.1
Router(config-subif)# xconnect 10.10.10.1 100 encapsulation mpls
Router(config-subif-xconn)# mtu ?
<64 - 1500> MTU size in bytes
Router(config-subif-xconn)# mtu 1501 <<================
Router(config-subif)# mtu ?
<64 - 17940> MTU size in bytes
```

If the MTU value is not accepted in either xconnect subinterface configuration mode or subinterface configuration mode, then the command is rejected.

Frame Relay over MPLS and DTE DCE and NNI Connections

You can configure an interface as a DTE device or a DCE switch, or as a switch connected to a switch with network-to-network interface (NNI) connections. Use the following command in interface configuration mode:

frame-relay intf-type [**dce** | **dte** | **nni**]

The keywords are explained in the table below.

Table 2: frame-relay intf-type Command Keywords

Keyword	Description
dce	Enables the router or access server to function as a switch connected to a router.
dte	Enables the router or access server to function as a DTE device. DTE is the default.
nni	Enables the router or access server to function as a switch connected to a switch.

Local Management Interface and Frame Relay over MPLS

Local Management Interface (LMI) is a protocol that communicates status information about PVCs. When a PVC is added, deleted, or changed, the LMI notifies the endpoint of the status change. LMI also provides a polling mechanism that verifies that a link is up.

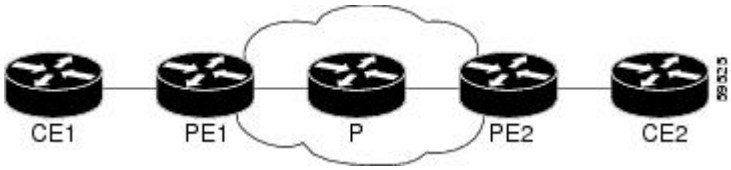
How LMI Works

To determine the PVC status, LMI checks that a PVC is available from the reporting device to the Frame Relay end-user device. If a PVC is available, LMI reports that the status is "Active," which means that all interfaces, line protocols, and core segments are operational between the reporting device and the Frame Relay end-user device. If any of those components is not available, the LMI reports a status of "Inactive."

Note Only the DCE and NNI interface types can report the LMI status.

The figure below is a sample topology that helps illustrate how LMI works.

Figure 1: Sample Topology



In the figure above, note the following:

- CE1 and PE1 and PE2 and CE2 are Frame Relay LMI peers.
- CE1 and CE2 can be Frame Relay switches or end-user devices.
- Each Frame Relay PVC comprises multiple segments.
- The DLCI value is local to each segment and is changed as traffic is switched from segment to segment. Two Frame Relay PVC segments exist in the figure; one is between PE1 and CE1 and the other is between PE2 and CE2.

The LMI protocol behavior depends on whether you have DLCI-to-DLCI or port-to-port connections.

DLCI-to-DLCI Connections

If you have DLCI-to-DLCI connections, LMI runs locally on the Frame Relay ports between the PE and CE devices:

- CE1 sends an active status to PE1 if the PVC for CE1 is available. If CE1 is a switch, LMI checks that the PVC is available from CE1 to the user device attached to CE1.
- PE1 sends an active status to CE1 if the following conditions are met:
  - A PVC for PE1 is available.
  - PE1 received an MPLS label from the remote PE router.

- An MPLS tunnel label exists between PE1 and the remote PE.

For DTE or DCE configurations, the following LMI behavior exists: The Frame Relay device accessing the network (DTE) does not report the PVC status. Only the network device (DCE) or NNI can report the status. Therefore, if a problem exists on the DTE side, the DCE is not aware of the problem.

Port-to-Port Connections

If you have port-to-port connections, the PE routers do not participate in the LMI status-checking procedures. LMI operates only between the CE routers. The CE routers must be configured as DCE-DTE or NNI-NNI.

For information about LMI, including configuration instructions, see the "Configuring the LMI" section of the Configuring Frame Relay document.

QoS Features Supported with AToM

For information about configuring QoS features on Cisco 12000 series routers, see the following feature module:

Any Transport over MPLS (AToM): Layer 2 QoS for the Cisco 12000 Series Router (Quality of Service)

The tables below list the QoS features supported by AToM on the Cisco 7200 and 7500 series routers.

Table 3: QoS Features Supported with Ethernet over MPLS on the Cisco 7200 and 7500 Series Routers

QoS Feature	Ethernet over MPLS
Service policy	Can be applied to: • Interface (input and output) • Subinterface (input and output)
Classification	Supports the following commands: • **match cos** (on interfaces and subinterfaces) • **match mpls experimental** (on interfaces and subinterfaces) • **match qos-group** (on interfaces) (output policy)
Marking	Supports the following commands: • **set cos** (output policy) • **set discard-class** (input policy) • **set mpls experimental** (input policy) (on interfaces and subinterfaces) • **set qos-group** (input policy)

QoS Feature	Ethernet over MPLS
Policing	Supports the following: • Single-rate policing • Two-rate policing • Color-aware policing • Multiple-action policing
Queueing and shaping	Supports the following: • Distributed Low Latency Queueing (dLLQ) • Distributed Weighted Random Early Detection (dWRED) • Byte-based WRED

Table 4: QoS Features Supported with Frame Relay over MPLS on the Cisco 7200 and 7500 Series Routers

QoS Feature	Frame Relay over MPLS
Service policy	Can be applied to: • Interface (input and output) • PVC (input and output)
Classification	Supports the following commands: • **match fr-de** (on interfaces and VCs) • **match fr-dlci** (on interfaces) • **match qos-group**
Marking	Supports the following commands: • **frame-relay congestion management** (output) • **set discard-class** • **set fr-de** (output policy) • **set fr-fecn-becn** (output) • **set mpls experimental** • **set qos-group** • **threshold ecn** (output)

QoS Feature	Frame Relay over MPLS
Policing	Supports the following: • Single-rate policing • Two-rate policing • Color-aware policing • Multiple-action policing
Queueing and shaping	Supports the following: • dLLQ • dWRED • Distributed traffic shaping • Distributed class-based weighted fair queueing (dCBWFQ) • Byte-based WRED • **random-detect discard-class-based** command

Table 5: QoS Features Supported with ATM Cell Relay and AAL5 over MPLS on the Cisco 7200 and 7500 Series Routers

QoS Feature	ATM Cell Relay and AAL5 over MPLS
Service policy	Can be applied to: • Interface (input and output) • Subinterface (input and output) • PVC (input and output)
Classification	Supports the following commands: • **match mpls experimental** (on VCs) • **match qos-group** (output)

QoS Feature	ATM Cell Relay and AAL5 over MPLS
Marking	Supports the following commands: • **random-detect discard-class-based** (input) • **set clp** (output) (on interfaces, subinterfaces, and VCs) • **set discard-class** (input) • **set mpls experimental** (input) (on interfaces, subinterfaces, and VCs) • **set qos-group** (input)
Policing	Supports the following: • Single-rate policing • Two-rate policing • Color-aware policing • Multiple-action policing
Queueing and shaping	Supports the following: • dLLQ • dWRED • dCBWFQ • Byte-based WRED • random-detect discard-class-based command • Class-based shaping support on ATM PVCs

How to Configure Any Transport over MPLS

This section explains how to perform a basic AToM configuration and includes the following procedures:

Configuring the Pseudowire Class

The successful transmission of the Layer 2 frames between PE routers is due to the configuration of the PE routers. You set up the connection, called a pseudowire, between the routers.

Note In simple configurations, this task is optional. You do not need to specify a pseudowire class if you specify the tunneling method as part of the **xconnect**command.

The pseudowire-class configuration group specifies the following characteristics of the tunneling mechanism:

- Encapsulation type
- Control protocol
- Payload-specific options

For more information about the **pseudowire-class**command, see the following feature module: Layer 2 Tunnel Protocol Version 3.

You must specify the **encapsulation mpls**commandas part of the pseudowire class or as part of the **xconnect** command for the AToM VCs to work properly. If you omit the **encapsulation mpls**commandas part of the **xconnect**command, you will receive the following error:

```
% Incomplete command.
```

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *name*
4. **encapsulation mpls**
5. **end**

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** *name* **Example:** `Router(config)# pseudowire-class atom`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.

	Command or Action	Purpose
Step 4	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 5	**end** **Example:** `Router(config-pw-class)# end`	Exits pseudowire class configuration mode and returns to privileged EXEC mode.

What to Do Next

To change the type of encapsulation, remove the pseudowire with the **no pseudowire-class** command, reestablish the pseudowire, and specify the new encapsulation type.

Once you specify the **encapsulation mpls** command, you can neither remove it using the **no encapsulation mpls**command nor change the command setting using the **encapsulation l2tpv3** command. If you try to remove or change the encapsulation type using the above-mentioned commands, you will get the following error message:

```
Encapsulation changes are not allowed on an existing pw-class.
```

To remove a pseudowire, use the **clear xconnect** command in privileged EXEC mode. You can remove all pseudowires or specific pseudowires on an interface or peer router.

Configuring ATM AAL5 over MPLS on PVCs

ATM AAL5 over MPLS for PVCs encapsulates ATM AAL5 service data unit (SDUs) in MPLS packets and forwards them across the MPLS network. Each ATM AAL5 SDU is transported as a single packet.

Note AAL5 over MPLS is supported only in SDU mode.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *typeslot/port*
4. **pvc** [*name*] *vpi/vci* **l2transport**
5. **encapsulation aal5**
6. **xconnect** *peer-router-id vcid* **encapsulation mpls**
7. **exit**
8. **exit**
9. **exit**
10. **show mpls l2transport vc**

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** *typeslot/port* **Example:** `Router(config)# interface atm1/0`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	**pvc** [*name*] *vpi/vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 5	**encapsulation aal5** **Example:** `Router(config-if-atm-l2trans-pvc)# encapsulation aal5`	Specifies the ATM ALL5 encapsulation for the PVC. • Make sure that you specify the same encapsulation type on the PE and CE routers.

	Command or Action	Purpose
Step 6	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.
Step 7	**exit** **Example:** `Router(config-if-atm-l2trans-pvc)# exit`	Exits L2transport PVC configuration mode.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 10	**show mpls l2transport vc** **Example:** `Router# show mpls l2transport vc`	Displays output that shows ATM AAL5 over MPLS is configured on a PVC.

Examples

The following is sample output from the **show mpls l2transport vc**command, which shows that ATM AAL5 over MPLS is configured on a PVC:

```
Router# show mpls l2transport vc
Local intf   Local circuit          Dest address      VC ID      Status
---------    -------------          ------------      -----      ------
ATM1/0       ATM AAL5 1/100         10.4.4.4           100        UP
```

Configuring ATM AAL5 over MPLS in VC Class Configuration Mode

You can create a VC class that specifies the AAL5 encapsulation and then attach the encapsulation type to an interface, subinterface, or PVC. The following task creates a VC class and attaches it to a main interface.

Note AAL5 over MPLS is supported only in SDU mode.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vc-class atm** *vc-class-name*
4. **encapsulation** *layer-type*
5. **exit**
6. **interface** *typeslot/port*
7. **class-int** *vc-class-name*
8. **pvc** [*name*] *vpi/vci* **l2transport**
9. **xconnect** *peer-router-id vcid* **encapsulation mpls**
10. **exit**
11. **exit**
12. **exit**
13. **show atm class-links**

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	**vc-class atm** *vc-class-name* **Example:** `Router(config)# vc-class atm aal5class`	Creates a VC class and enters VC class configuration mode.
Step 4	**encapsulation** *layer-type* **Example:** `Router(config-vc-class)# encapsulation aal5`	Configures AAL and the encapsulation type.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-vc-class)# exit`	Exits VC class configuration mode.
Step 6	**interface** *typeslot/port* **Example:** `Router(config)# interface atm1/0`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 7	**class-int** *vc-class-name* **Example:** `Router(config-if)# class-int aal5class`	Applies a VC class to the ATM main interface or subinterface. **Note** You can also apply a VC class to a PVC.
Step 8	**pvc** [*name*] *vpi/vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 9	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.
Step 10	**exit** **Example:** `Router(config-if-atm-l2trans-pvc)# exit`	Exits L2transport PVC configuration mode.
Step 11	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 12	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

	Command or Action	Purpose
Step 13	**show atm class-links** **Example:** `Router# show atm class-links`	Shows the type of encapsulation and that the VC class was applied to an interface.

Examples

In the following example, the command output of the **show atm class-links**command verifies that ATM AAL5 over MPLS is configured as part of a VC class. The command output shows the type of encapsulation and that the VC class was applied to an interface.

```
Router# show atm class-links 1/100
Displaying vc-class inheritance for ATM1/
0.0, vc 1/
100:
no broadcast - Not configured - using default
encapsulation aal5 - VC-class configured on main interface
```

Configuring OAM Cell Emulation for ATM AAL5 over MPLS

If a PE router does not support the transport of Operation, Administration, and Maintenance (OAM) cells across a label switched path (LSP), you can use OAM cell emulation to locally terminate or loop back the OAM cells. You configure OAM cell emulation on both PE routers, which emulates a VC by forming two unidirectional LSPs. You use the **oam-ac emulation-enable**and **oam-pvc manage**commands on both PE routers to enable OAM cell emulation.

After you enable OAM cell emulation on a router, you can configure and manage the ATM VC in the same manner as you would a terminated VC. A VC that has been configured with OAM cell emulation can send loopback cells at configured intervals toward the local CE router. The endpoint can be either of the following:

- End-to-end loopback, which sends OAM cells to the local CE router.
- Segment loopback, which responds to OAM cells to a device along the path between the PE and CE routers.

The OAM cells include the following cells:

- Alarm indication signal (AIS)
- Remote defect indication (RDI)

These cells identify and report defects along a VC. When a physical link or interface failure occurs, intermediate nodes insert OAM AIS cells into all the downstream devices affected by the failure. When a router receives an AIS cell, it marks the ATM VC down and sends an RDI cell to let the remote end know about the failure.

This section contains two tasks:

Configuring OAM Cell Emulation for ATM AAL5 over MPLS on PVCs

Perform this task to configure OAM cell emulation for ATM AAL5 over MPLS on a PVC.

Note For AAL5 over MPLS, you can configure the **oam-pvc manage**commandonly after you issue the **oam-ac emulation-enable** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *typeslot* /port
4. **pvc** [*name*] *vpi*/*vci* **l2transport**
5. **encapsulation aal5**
6. **xconnect** *peer-router-id vcid* **encapsulation mpls**
7. **oam-ac emulation-enable** [*ais-rate*]
8. **oam-pvc manage** [*frequency*]
9. **exit**
10. **exit**
11. **exit**
12. **show atm pvc**

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** *typeslot* /port **Example:** `Router(config)# interface atm1/0`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**pvc** [*name*] *vpi*/*vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 5	**encapsulation aal5** **Example:** `Router(config-if-atm-l2trans-pvc)# encapsulation aal5`	Specifies ATM AAL5 encapsulation for the PVC. • Make sure you specify the same encapsulation type on the PE and CE routers.
Step 6	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.
Step 7	**oam-ac emulation-enable** [*ais-rate*] **Example:** `Router(config-if-atm-l2trans-pvc)# oam-ac emulation-enable 30`	Enables OAM cell emulation for AAL5 over MPLS. • The *ais-rate* argument lets you specify the rate at which AIS cells are sent. The default is one cell every second. The range is 0 to 60 seconds.
Step 8	**oam-pvc manage** [*frequency*] **Example:** `Router(config-if-atm-l2trans-pvc)# oam-pvc manage`	Enables the PVC to generate end-to-end OAM loopback cells that verify connectivity on the virtual circuit. • The optional *frequency* argument is the interval between transmission of loopback cells and ranges from 0 to 600 seconds. The default value is 10 seconds.
Step 9	**exit** **Example:** `Router(config-if-atm-l2trans-pvc)# exit`	Exits L2transport PVC configuration mode.
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 11	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

	Command or Action	Purpose
Step 12	**show atm pvc** **Example:** `Router# show atm pvc`	Displays output that shows OAM cell emulation is enabled on the ATM PVC.

Examples

The output of the **show atm pvc** command in the following example shows that OAM cell emulation is enabled on the ATM PVC:

```
Router# show atm pvc 5/500
ATM4/1/0.200: VCD: 6, VPI: 5, VCI: 500
UBR, PeakRate: 1
AAL5-LLC/SNAP, etype:0x0, Flags: 0x34000C20, VCmode: 0x0
OAM Cell Emulation: enabled, F5 End2end AIS Xmit frequency: 1 second(s)
OAM frequency: 0 second(s), OAM retry frequency: 1 second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Disabled
OAM VC state: Not ManagedVerified
ILMI VC state: Not Managed
InPkts: 564, OutPkts: 560, InBytes: 19792, OutBytes: 19680
InPRoc: 0, OutPRoc: 0
InFast: 4, OutFast: 0, InAS: 560, OutAS: 560
InPktDrops: 0, OutPktDrops: 0
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0
Out CLP=1 Pkts: 0
OAM cells received: 26
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 26
OAM cells sent: 77
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutAIS: 77, F5 OutRDI: 0
OAM cell drops: 0
Status: UP
```

Configuring OAM Cell Emulation for ATM AAL5 over MPLS in VC Class Configuration Mode

The following steps explain how to configure OAM cell emulation as part of a VC class. You can then apply the VC class to an interface, a subinterface, or a VC. When you configure OAM cell emulation in VC class configuration mode and then apply the VC class to an interface, the settings in the VC class apply to all the VCs on the interface, unless you specify a different OAM cell emulation value at a lower level, such as the subinterface or VC level. For example, you can create a VC class that specifies OAM cell emulation and sets the rate of AIS cells to every 30 seconds. You can apply the VC class to an interface. Then, for one PVC, you can enable OAM cell emulation and set the rate of AIS cells to every 15 seconds. All the PVCs on the interface use the cell rate of 30 seconds, except for the one PVC that was set to 15 seconds.

Perform this task to enable OAM cell emulation as part of a VC class and apply it to an interface.

Note For AAL5 over MPLS, you can configure the **oam-pvc manage**command only after you issue the **oam-ac emulation-enable** command**.**

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vc-class atm** *name*
4. **encapsulation** *layer-type*
5. **oam-ac emulation-enable** [*ais-rate*]
6. **oam-pvc manage** [*frequency*]
7. **exit**
8. **interface** *typeslot/port*
9. **class-int** *vc-class-name*
10. **pvc** [*name*] *vpi/vci* **l2transport**
11. **xconnect** *peer-router-id vcid* **encapsulation mpls**
12. **end**

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	**vc-class atm** *name* **Example:** `Router(config)# vc-class atm oamclass`	Creates a VC class and enters VC class configuration mode.
Step 4	**encapsulation** *layer-type* **Example:** `Router(config-vc-class)# encapsulation aal5`	Configures the AAL and encapsulation type.
Step 5	**oam-ac emulation-enable** [*ais-rate*] **Example:** `Router(config-vc-class)# oam-ac emulation-enable 30`	Enables OAM cell emulation for AAL5 over MPLS. • The *ais-rate* argument lets you specify the rate at which AIS cells are sent. The default is one cell every second. The range is 0 to 60 seconds.

	Command or Action	Purpose
Step 6	**oam-pvc manage** [*frequency*] **Example:** `Router(config-vc-class)# oam-pvc manage`	Enables the PVC to generate end-to-end OAM loopback cells that verify connectivity on the virtual circuit. • The optional *frequency* argument is the interval between transmission of loopback cells and ranges from 0 to 600 seconds. The default value is 10 seconds.
Step 7	**exit** **Example:** `Router(config-vc-class)# exit`	Exits VC class configuration mode.
Step 8	**interface** *typeslot/port* **Example:** `Router(config)# interface atm1/0`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 9	**class-int** *vc-class-name* **Example:** `Router(config-if)# class-int oamclass`	Applies a VC class to the ATM main interface or subinterface. **Note** You can also apply a VC class to a PVC.
Step 10	**pvc** [*name*] *vpi/vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 11	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.
Step 12	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits L2transport PVC configuration mode and returns to privileged EXEC mode.

Configuring ATM Cell Relay over MPLS in VC Mode

Perform this task to configure ATM cell relay on the permanent virtual circuits.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot* /*port*
4. **pvc** *vpi/vci* **l2transport**
5. **encapsulation aal0**
6. **xconnect** *peer-router-id vcid* **encapsulation mpls**
7. **exit**
8. **exit**
9. **exit**
10. **show atm vc**

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 atm** *slot* /*port* **Example:** `Router(config)# interface atm1/0`	Specifies an ATM interface and enters interface configuration mode.
Step 4	**pvc** *vpi/vci* **l2transport** **Example:** `Router(config-if)# pvc 0/100 l2transport`	Assigns a virtual path identifier (VPI) and virtual circuit identifier (VCI) and enters L2transport PVC configuration mode. • The **l2transport**keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 5	**encapsulation aal0** **Example:** `Router(config-if-atm-l2trans-pvc)# encapsulation aal0`	For ATM cell relay, specifies raw cell encapsulation for the interface. • Make sure you specify the same encapsulation type on the PE and CE routers.

	Command or Action	Purpose
Step 6	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.
Step 7	**exit** **Example:** `Router(config-if-atm-l2trans-pvc)# exit`	Exits L2transport PVC configuration mode.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 10	**show atm vc** **Example:** `Router# show atm vc`	Verifies that OAM cell emulation is enabled on the ATM VC.

Examples

The output of the **show atm vc** command shows that the interface is configured for VC mode cell relay:

```
Router# show atm vc 7
ATM3/0: VCD: 7, VPI: 23, VCI: 100
UBR, PeakRate: 149760
AAL0-Cell Relay, etype:0x10, Flags: 0x10000C2D, VCmode: 0x0
OAM Cell Emulation: not configured
InBytes: 0, OutBytes: 0
Status: UP
```

Configuring ATM Cell Relay over MPLS in VC Mode Using VC Class Configuration Mode

You can create a VC class that specifies the ATM cell relay encapsulation and then attach the VC class to an interface, subinterface, or VC. The following task creates a VC class that specifies the ATM cell relay encapsulation and attaches it to a main interface.

Note You can configure VC class configuration mode only in VC mode. VC class configuration mode is not supported on VP or port mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vc-class atm** *name*
4. **encapsulation** *layer-type*
5. **exit**
6. **interface** *typeslot /port*
7. **class-int** *vc-class-name*
8. **pvc** [*name*] *vpi/vci* **l2transport**
9. **xconnect** *peer-router-id vcid* **encapsulation mpls**
10. **end**

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	**vc-class atm** *name* **Example:** `Router(config)# vc-class atm cellrelay`	Creates a VC class and enters VC class configuration mode.
Step 4	**encapsulation** *layer-type* **Example:** `Router(config-vc-class)# encapsulation aal0`	Configures the AAL and encapsulation type.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-vc-class)# exit`	Exits VC class configuration mode.
Step 6	**interface** *typeslot /port* **Example:** `Router(config)# interface atm1/0`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 7	**class-int** *vc-class-name* **Example:** `Router(config-if)# class-int cellrelay`	Applies a VC class to the ATM main interface or subinterface. **Note** You can also apply a VC class to a PVC.
Step 8	**pvc** [*name*] *vpi*/*vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 9	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.
Step 10	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits L2transport PVC configuration mode and returns to privileged EXEC mode.

Configuring ATM Cell Relay over MPLS in PVP Mode

VP mode allows cells coming into a predefined PVP on the ATM interface to be transported over the MPLS backbone to a predefined PVP on the egress ATM interface. You can use VP mode to send single cells or packed cells over the MPLS backbone.

To configure VP mode, you must specify the following:

- The VP for transporting cell relay cells.
- The IP address of the peer PE router and the VC ID.

When configuring ATM cell relay over MPLS in VP mode, use the following guidelines:

- You do not need to enter the **encapsulation aal0** command in VP mode.
- One ATM interface can accommodate multiple types of ATM connections. VP cell relay, VC cell relay, and ATM AAL5 over MPLS can coexist on one ATM interface. On the Cisco 12000 series router, this is true only on the engine 0 ATM line cards.
- If a VPI is configured for VP cell relay, you cannot configure a PVC using the same VPI.
- VP trunking (mapping multiple VPs to one emulated VC label) is not supported. Each VP is mapped to one emulated VC.
- Each VP is associated with one unique emulated VC ID. The AToM emulated VC type is ATM VP cell transport.
- The AToM control word is supported. However, if a peer PE does not support the control word, it is disabled. This negotiation is done by LDP label binding.
- VP mode (and VC mode) drop idle cells.

Perform this task to configure ATM cell relay in PVP mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot /port*
4. **atm pvp** *vpi* **l2transport**
5. **xconnect** *peer-router-id vcid* **encapsulation mpls**
6. **exit**
7. **exit**
8. **exit**
9. **show atm vp**

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	**interface atm** *slot /port* **Example:** `Router(config)# interface atm1/0`	Defines the interface and enters interface configuration mode.
Step 4	**atm pvp** *vpi* **l2transport** **Example:** `Router(config-if)# atm pvp 1 l2transport`	Specifies that the PVP is dedicated to transporting ATM cells and enters L2transport PVP configuration mode. • The **l2transport** keyword indicates that the PVP is for cell relay. This mode is for Layer 2 transport only; it is not for regular PVPs.
Step 5	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvp)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. • The syntax for this command is the same as for all other Layer 2 transports.
Step 6	**exit** **Example:** `Router(config-if-atm-l2trans-pvP)# exit`	Exits L2 transport PVP configuration mode.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 8	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 9	**show atm vp** **Example:** `Router# show atm vp`	Displays output that shows OAM cell emulation is enabled on the ATM VP.

Examples

The following **show atm vp** command in the following example shows that the interface is configured for VP mode cell relay:

```
Router# show atm vp 1
```

```
ATM5/0  VPI: 1, Cell Relay, PeakRate: 149760, CesRate: 0, DataVCs: 1, CesVCs: 0, Status:
ACTIVE
  VCD    VCI   Type   InPkts   OutPkts   AAL/Encap     Status
  6      3     PVC    0        0         F4 OAM        ACTIVE
  7      4     PVC    0        0         F4 OAM        ACTIVE
TotalInPkts: 0, TotalOutPkts: 0, TotalInFast: 0, TotalOutFast: 0,
TotalBroadcasts: 0 TotalInPktDrops: 0, TotalOutPktDrops: 0
```

Configuring ATM Cell Relay over MPLS in Port Mode

Port mode cell relay allows cells coming into an ATM interface to be packed into an MPLS packet and transported over the MPLS backbone to an egress ATM interface.

To configure port mode, issue the **xconnect** command from an ATM main interface and specify the destination address and the VC ID. The syntax of the **xconnect** command is the same as for all other transport types. Each ATM port is associated with one unique pseudowire VC label.

When configuring ATM cell relay over MPLS in port mode, use the following guidelines:

- The pseudowire VC type is set to ATM transparent cell transport (AAL0).
- The AToM control word is supported. However, if the peer PE does not support a control word, the control word is disabled. This negotiation is done by LDP label binding.

Note The AToM control word is not supported for port mode cell relay on Cisco 7600 series routers.

- Port mode and VP and VC mode are mutually exclusive. If you enable an ATM main interface for cell relay, you cannot enter any PVP or PVC commands.
- If the pseudowire VC label is withdrawn due to an MPLS core network failure, the PE router sends a line AIS to the CE router.
- For the Cisco 7600 series routers, you must specify the interface ATM slot, bay, and port for the SIP400 or SIP200.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot /port*
4. **xconnect** *peer-router-id vcid* **encapsulation mpls**
5. **exit**
6. **exit**
7. **show atm route**
8. **show mpls l2transport vc**

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 atm** *slot /port*
	Example:
	`or` **interface atm** *slot/bay/port*
	Example:
	`Router(config)# interface atm1/0`
	Example:
	`or`
	Example:
	`Router(config)# interface atm4/3/0`
		Specifies an ATM interface and enters interface configuration mode.
		• For the Cisco 7600 series routers, you must specify the interface ATM slot, bay, and port for the SIP400 or SIP200. In the example the slot is 4, the bay is 3, and the port is 0.

Step 4	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to the interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

	Command or Action	Purpose
Step 7	**show atm route** **Example:** `Router# show atm route`	Displays output that shows ATM cell relay in port mode has been enabled.
Step 8	**show mpls l2transport vc** **Example:** `Router# show mpls l2transport vc`	Displays the attachment circuit and the interface.

Examples

The **show atm route** command in the following example displays port mode cell relay state. The following example shows that atm interface 1/0 is for cell relay, the VC ID is 123 and the tunnel is down.

```
Router# show atm route
Input Intf        Output Intf     Output VC       Status
ATM1/0            ATOM Tunnel     123             DOWN
```

The **show mpls l2transport vc** command in the following example also shows configuration information:

```
Router# show mpls l2transport vc
Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
AT1/0          ATM CELL ATM1/0      10.1.1.121      1121       UP
```

Troubleshooting Tips

The **debug atm l2transport** and **debug mpls l2transport vc**display troubleshooting information.

Configuring ATM Single Cell Relay over MPLS

The single cell relay feature allows you to insert one ATM cell in each MPLS packet. You can use single cell relay in both VP and VC mode. The configuration steps show how to configure single cell relay in VC mode. For VP mode, see the Configuring ATM Cell Relay over MPLS in PVP Mode, on page 30.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot/port*
4. **pvc** *vpi/vci* **l2transport**
5. **encapsulation aal0**
6. **xconnect** *peer-router-id vcid* **encapsulation mpls**
7. **end**

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 atm** *slot/port* **Example:** `Router(config)# interface atm1/0`	Specifies an ATM interface and enters interface configuration mode.
Step 4	**pvc** *vpi/vci* **l2transport** **Example:** `Router(config-if)# pvc 1/100 l2transport`	Assigns a VPI and VCI and enters L2transport PVC configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 5	**encapsulation aal0** **Example:** `Router(config-if-atm-l2trans-pvc)# encapsulation aal0`	Specifies raw cell encapsulation for the interface. • Make sure you specify the same encapsulation type on the PE and CE routers.
Step 6	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.

	Command or Action	Purpose
Step 7	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits L2transport PVC configuration mode and returns to privileged EXEC mode.

Configuring ATM Packed Cell Relay over MPLS

The packed cell relay feature allows you to insert multiple concatenated ATM cells in an MPLS packet. The packed cell relay feature is more efficient than single cell relay, because each ATM cell is 52 bytes, and each AToM packet is at least 64 bytes.

At a high level, packed cell relay configuration consists of the following steps:

1. You specify the amount of time a PE router can wait for cells to be packed into an MPLS packet. You can set up three timers by default with different amounts of time attributed to each timer.
2. You enable packed cell relay, specify how many cells should be packed into each MPLS packet, and choose which timer to use during the cell packing process.

Restrictions

- The **cell-packing**command is available only if you use AAL0 encapsulation in VC mode. If the command is configured with ATM AAL5 encapsulation, the command is not valid.
- Only cells from the same VC, VP, or port can be packed into one MPLS packet. Cells from different connections cannot be concatenated into the same MPLS packet.
- When you change, enable, or disable the cell-packing attributes, the ATM VC, VP, or port and the MPLS emulated VC are reestablished.
- If a PE router does not support packed cell relay, the PE router sends only one cell per MPLS packet.
- The number of packed cells does not need to match between the PE routers. The two PE routers agree on the lower of the two values. For example, if PE1 is allowed to pack 10 cells per MPLS packet and PE2 is allowed to pack 20 cells per MPLS packet, the two PE routers would agree to send no more than 10 cells per packet.
- If the number of cells packed by the peer PE router exceeds the limit, the packet is dropped.
- Issue the **atm mcpt-timers**command on an ATM interface before issuing the **cell-packing**command.

See the following sections for configuration information:

Configuring ATM Packed Cell Relay over MPLS in VC Mode

Perform this task to configure the ATM packed cell relay over MPLS feature in VC mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot /port*
4. **shutdown**
5. **atm mcpt-timers** [*timer1-timeout timer2-timeout timer3-timeout*]
6. **no shutdown**
7. **pvc** *vpi/vci* **l2transport**
8. **encapsulation aal0**
9. **xconnect** *peer-router-id vcid* **encapsulation mpls**
10. **cell-packing** *cells* **mcpt-timer** *timer*
11. **end**

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 atm** *slot /port* **Example:** `Router(config)# interface atm1/0`	Defines the interface and enters interface configuration mode.
Step 4	**shutdown** **Example:** `Router(config-if)# shutdown`	Shuts down the interface.
Step 5	**atm mcpt-timers** [*timer1-timeout timer2-timeout timer3-timeout*] **Example:** `Router(config-if)# atm mcpt-timers 100 200 250`	Sets up the cell-packing timers, which specify how long the PE router can wait for cells to be packed into an MPLS packet. • You can set up to three timers. For each timer, you specify the maximum cell-packing timeout (MCPT). This value gives the cell-packing function a limited amount of time to complete. If the timer expires before the maximum number of cells are packed into

	Command or Action	Purpose

	Example:	an AToM packet, the packet is sent anyway. The timeout's default and range of acceptable values depends on the ATM link speed.
		• The respective default values for the PA-A3 port adapters are:
		  • OC-3: 30, 60, and 90 microseconds
		  • T3: 100, 200, and 300 microseconds
		  • E3: 130, 260, and 390 microseconds
		• You can specify either the number of microseconds or use the default.
		• The respective range of values for the PA-A3 port adapters are:
		  • OC-3: 10 to 4095 microseconds
		  • T3: 30 to 4095 microseconds
		  • E3: 40 to 4095 microseconds

Step 6	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 7	**pvc** *vpi/vci* **l2transport** **Example:** `Router(config-if)# pvc 1/100 l2transport`	Assigns a VPI and VCI and enters L2transport PVC configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 8	**encapsulation aal0** **Example:** `Router(config-if-atm-l2trans-pvc)# encapsulation aal0`	Specifies raw cell encapsulation for the interface. • Make sure you specify the same encapsulation type on the PE routers.
Step 9	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.
Step 10	**cell-packing** *cells* **mcpt-timer** *timer*	Enables cell packing and specifies the cell-packing parameters.

	Command or Action	Purpose
	Example: `Router(config-if-atm-l2trans-pvc)# cell-packing 10 mcpt-timer 1` **Example:**	• The *cells*argument represents the maximum number of cells to be packed into an MPLS packet. The range is from 2 to the MTU of the interface divided by 52. The default is MTU/52. • The *timer*argument allows you to specify which timer to use. The default is timer 1. • See the **cell-packing** command page for more information.
Step 11	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits L2transport PVC configuration mode and returns to privileged EXEC mode.

Configuring ATM Packed Cell Relay over MPLS in VC Mode Using VC Class Configuration Mode

You can create a VC class that specifies the ATM cell relay encapsulation and the cell packing parameters and then attach the VC class to an interface, subinterface, or VC. The following task creates a VC class that specifies the ATM cell relay encapsulation and cell packing and attaches it to a main interface.

Note You can configure VC class configuration mode only in VC mode. VC class configuration mode is not supported on VP or port mode.

When you configure cell packing in VC class configuration mode and then apply the VC class to an interface, the settings in the VC class apply to all the VCs on the interface, unless you specify a different cell packing value at a lower level, such as the subinterface or VC level. For example, you can create a VC class that specifies three cells to be packed. You can apply the VC class to an interface. Then, for one PVC, you can specify two cells to be packed. All the PVCs on the interface pack three cells, except for the one PVC that was set to set two cells.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vc-class atm** *name*
4. **encapsulation** *layer-type*
5. **cell-packing** *cells* **mcpt-timer** *timer*
6. **exit**
7. **interface** *typeslot /port*
8. **shutdown**
9. **atm mcpt-timers** [*timer1-timeout timer2-timeout timer3-timeout*]
10. **no shutdown**
11. **class-int** *vc-class-name*
12. **pvc** [*name*] *vpi*/*vci* **l2transport**
13. **xconnect** *peer-router-id vcid* **encapsulation mpls**
14. **end**

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	**vc-class atm** *name* **Example:** `Router(config)# vc-class atm cellpacking`	Creates a VC class and enters VC class configuration mode.
Step 4	**encapsulation** *layer-type* **Example:** `Router(config-vc-class)# encapsulation aal0`	Configures the AAL and encapsulation type.
Step 5	**cell-packing** *cells* **mcpt-timer** *timer*	Enables cell packing and specifies the cell-packing parameters.

	Command or Action	Purpose
	Example: `Router(config-vc-class)# cell-packing 10 mcpt-timer 1` **Example:**	• The *cells*argument represents the maximum number of cells to be packed into an MPLS packet. The range is from 2 to the MTU of the interface divided by 52. The default is MTU/52. • The *timer*argument allows you to specify which timer to use. The default is timer 1. • See the **cell-packing** command page for more information.
Step 6	**exit** **Example:** `Router(config-vc-class)# exit`	Exits VC class configuration mode.
Step 7	**interface** *typeslot /port* **Example:** `Router(config)# interface atm1/0`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 8	**shutdown** **Example:** `Router(config-if)# shutdown`	Shuts down the interface.

Step 9	**atm mcpt-timers** [*timer1-timeout timer2-timeout timer3-timeout*]
	Example:
	`Router(config-if)# atm mcpt-timers 100 200 250`
	Example:
		Sets up the cell-packing timers, which specify how long the PE router can wait for cells to be packed into an MPLS packet.
		• You can set up to three timers. For each timer, you specify the MCPT. This value gives the cell-packing function a limited amount of time to complete. If the timer expires before the maximum number of cells are packed into an AToM packet, the packet is sent anyway. The timeout's default and range of acceptable values depends on the ATM link speed.
		• The respective default values for the PA-A3 port adapters are:
		  • OC-3: 30, 60, and 90 microseconds
		  • T3: 100, 200, and 300 microseconds
		  • E3: 130, 260, and 390 microseconds
		• You can specify either the number of microseconds or use the default.
		• The respective range of values for the PA-A3 port adapters are:
		  • OC-3: 10 to 4095 microseconds
		  • T3: 30 to 4095 microseconds

	Command or Action	Purpose
		• E3: 40 to 4095 microseconds
Step 10	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 11	**class-int** *vc-class-name* **Example:** `Router(config-if)# class-int cellpacking`	Applies a VC class to the ATM main interface or subinterface. **Note** You can also apply a VC class to a PVC.
Step 12	**pvc** [*name*] *vpi*/*vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 13	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.
Step 14	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits L2transport PVC configuration mode and returns to privileged EXEC mode.

Configuring ATM Packed Cell Relay over MPLS in VP Mode

Perform this task to configure the ATM cell-packing feature in VP mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot /port*
4. **shutdown**
5. **atm mcpt-timers** [*timer1-timeout timer2-timeout timer3-timeout*]
6. **no shutdown**
7. **atm pvp** *vpi* **l2transport**
8. **xconnect** *peer-router-id vcid* **encapsulation mpls**
9. **cell-packing** *cells* **mcpt-timer** *timer*
10. **end**

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 atm** *slot /port* **Example:** `Router(config)# interface atm1/0`	Defines the interface and enters interface configuration mode.
Step 4	**shutdown** **Example:** `Router(config-if)# shutdown`	Shuts down the interface.
Step 5	**atm mcpt-timers** [*timer1-timeout timer2-timeout timer3-timeout*] **Example:** `Router(config-if)# atm mcpt-timers 100 200 250`	Sets up the cell-packing timers, which specify how long the PE router can wait for cells to be packed into an MPLS packet. • You can set up to three timers. For each timer, you specify the MCPT. This value gives the cell-packing function a limited amount of time to complete. If the timer expires before the maximum number of cells are packed into an AToM packet, the packet is sent anyway. The timeout's default and range of acceptable values depends on the ATM link speed.

	Command or Action	Purpose
	Example:	• The respective default values for the PA-A3 port adapters are: • OC-3: 30, 60, and 90 microseconds • T3: 100, 200, and 300 microseconds • E3: 130, 260, and 390 microseconds • You can specify either the number of microseconds or use the default. • The respective range of values for the PA-A3 port adapters are: • OC-3: 10 to 4095 microseconds • T3: 30 to 4095 microseconds • E3: 40 to 4095 microseconds
Step 6	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 7	**atm pvp** *vpi* **l2transport** **Example:** `Router(config-if)# atm pvp 1 l2transport`	Specifies that the PVP is dedicated to transporting ATM cells and enters L2transport PVP configuration mode. • The **l2transport** keyword indicates that the PVP is for cell relay. This mode is for Layer 2 transport only; it is not for regular PVPs.
Step 8	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(cfg-if-atm-l2trans-pvp)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. • The syntax for this command is the same as for all other Layer 2 transports.
Step 9	**cell-packing** *cells* **mcpt-timer** *timer* **Example:** `Router(cfg-if-atm-l2trans-pvp)# cell-packing 10 mcpt-timer 1` **Example:**	Enables cell packing and specifies the cell-packing parameters. • The *cells*argument represents the maximum number of cells to be packed into an MPLS packet. The range is from 2 to the MTU of the interface divided by 52. The default is MTU/52. • The *timer*argument allows you to specify which timer to use. The default is timer 1. • See the **cell-packing** command page for more information.

	Command or Action	Purpose
Step 10	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits L2transport PVC configuration mode and returns to privileged EXEC mode.

Configuring ATM Packed Cell Relay over MPLS in Port Mode

Perform this task to configure ATM packed cell relay over MPLS in port mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot /port*
4. **shutdown**
5. **atm mcpt-timers** [*timer1-timeout timer2-timeout timer3-timeout*]
6. **no shutdown**
7. **cell-packing** *cells* **mcpt-timer** *timer*
8. **xconnect** *peer-router-id vcid* **encapsulation mpls**
9. **exit**
10. **exit**
11. **show atm cell-packing**
12. **show atm vp**

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	**interface atm** *slot /port* **Example:** `Router(config)# interface atm1/0`	Specifies an ATM interface and enters interface configuration mode.
Step 4	**shutdown** **Example:** `Router(config-if)# shutdown`	Shuts down the interface.

Step 5	**atm mcpt-timers** [*timer1-timeout timer2-timeout timer3-timeout*]
	
	Example:
	
	`Router(config-if)# atm mcpt-timers 100 200 250`
		Sets up the cell-packing timers, which specify how long the PE router can wait for cells to be packed into an MPLS packet.
		• You can set up to three timers. For each timer, you specify the MCPT. This value gives the cell-packing function a limited amount of time to complete. If the timer expires before the maximum number of cells are packed into an AToM packet, the packet is sent anyway. The timeout's default and range of acceptable values depends on the ATM link speed.
		• The respective default values for the PA-A3 port adapters are:
		  • OC-3: 30, 60, and 90 microseconds
		  • T3: 100, 200, and 300 microseconds
		  • E3: 130, 260, and 390 microseconds
		• You can specify either the number of microseconds or use the default.
		• The respective range of values for the PA-A3 port adapters are:
		  • OC-3: 10 to 4095 microseconds
		  • T3: 30 to 4095 microseconds
		  • E3: 40 to 4095 microseconds

Step 6	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 7	**cell-packing** *cells* **mcpt-timer** *timer* **Example:** `Router(config-if)# cell-packing 10 mcpt-timer 1`	Enables cell packing and specifies the cell-packing parameters. • The *cells*argument represents the maximum number of cells to be packed into an MPLS packet. The range is from 2 to the MTU of the interface divided by 52. The default is MTU/52.

	Command or Action	Purpose
	Example:	• The *timer*argument allows you to specify which timer to use. The default is timer 1. • See the cell-packing command page for more information.
Step 8	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to the interface.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show atm cell-packing** **Example:** `Router# show atm cell-packing`	Displays cell-packing statistics.
Step 12	**show atm vp** **Example:** `Router# show atm vp`	Displays cell-packing information.

Examples

The **show atm cell-packing** command in the following example displays the following statistics:

- The number of cells that are to be packed into an MPLS packet on the local and peer routers
- The average number of cells sent and received
- The timer values associated with the local router

```
Router# show atm cell-packing
                      average                  average
        circuit  local  nbr of cells    peer    nbr of cells    MCPT
        type     MNCP   rcvd in one pkt MNCP    sent in one pkt (us)
```

```
================================================================================
atm 1/0 vc 1/200  20    15              30              20       60
atm 1/0 vp 2      25    21              30              24      100
```

The **show atm vp** command in the following example displays the cell packing information at the end of the output:

```
Router# show atm vp 12
ATM5/0  VPI: 12, Cell Relay, PeakRate: 149760, CesRate: 0, DataVCs: 1, CesVCs: 0, Status:
ACTIVE
  VCD    VCI   Type   InPkts   OutPkts   AAL/Encap     Status
  6      3     PVC    0        0         F4 OAM        ACTIVE
  7      4     PVC    0        0         F4 OAM        ACTIVE
TotalInPkts: 0, TotalOutPkts: 0, TotalInFast: 0, TotalOutFast: 0,
TotalBroadcasts: 0 TotalInPktDrops: 0, TotalOutPktDrops: 0
Local MNCP: 5, average number of cells received: 3
Peer MNCP: 1, average number of cells sent: 1
Local MCPT: 100 us
```

Troubleshooting Tips

To debug ATM cell packing, issue the **debug atm cell-packing** command.

Configuring Ethernet over MPLS in VLAN Mode

A VLAN is a switched network that is logically segmented by functions, project teams, or applications regardless of the physical location of users. Ethernet over MPLS allows you to connect two VLAN networks that are in different locations. You configure the PE routers at each end of the MPLS backbone and add a point-to-point VC. Only the two PE routers at the ingress and egress points of the MPLS backbone know about the VCs dedicated to transporting Layer 2 VLAN traffic. All other routers do not have table entries for those VCs. Ethernet over MPLS in VLAN mode transports Ethernet traffic from a source 802.1Q VLAN to a destination 802.1Q VLAN over a core MPLS network.

Note You must configure Ethernet over MPLS (VLAN mode) on the subinterfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot /interface.subinterface*
4. **encapsulation dot1q** *vlan-id*
5. **xconnect** *peer-router-id vcid* **encapsulation mpls**
6. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot /interface.subinterface* **Example:** `Router(config)# interface gigabitethernet4/0.1`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. • Make sure the subinterface on the adjoining CE router is on the same VLAN as this PE router.
Step 4	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-subif)# encapsulation dot1q 100`	Enables the subinterface to accept 802.1Q VLAN packets. • The subinterfaces between the CE and PE routers that are running Ethernet over MPLS must be in the same subnet. All other subinterfaces and backbone routers do not.
Step 5	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-subif)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. • The syntax for this command is the same as for all other Layer 2 transports.
Step 6	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits L2transport PVC configuration mode and returns to privileged EXEC mode.

Configuring Ethernet over MPLS in Port Mode

Port mode allows a frame coming into an interface to be packed into an MPLS packet and transported over the MPLS backbone to an egress interface. The entire Ethernet frame without the preamble or FCS is transported as a single packet. To configure port mode, use the **xconnect** command in interface configuration mode and specify the destination address and the VC ID. The syntax of the **xconnect** command is the same as for all other transport types. Each interface is associated with one unique pseudowire VC label.

When configuring Ethernet over MPLS in port mode, use the following guidelines:

- The pseudowire VC type is set to Ethernet.
- Port mode and Ethernet VLAN mode are mutually exclusive. If you enable a main interface for port-to-port transport, you cannot also enter commands on a subinterface.
- In Cisco IOS Release 12.2(33)SRE and later releases, L2VPN Routed Interworking using Ethernet over MPLS (EOMPLS) is no longer supported. When you configure the **interworking ip** command in pseudowire configuration mode, the **xconnect** command is disabled. To configure L2VPN Routed Interworking, use either Ethernet over MPLS (EOMPLS) or SVI (Switched Virtual Interface) based EOMPLS.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot/interface*
4. **xconnect** *peer-router-id vcid* **encapsulation mpls**
5. **exit**
6. **exit**
7. **show mpls l2transport vc**

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 gigabitethernet** *slot/interface* **Example:** `Router(config)# interface gigabitethernet4/0`	Specifies the Gigabit Ethernet interface and enters interface configuration mode. • Make sure the interface on the adjoining CE router is on the same VLAN as this PE router.
Step 4	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. • The syntax for this command is the same as for all other Layer 2 transports.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits router configuration mode.
Step 7	**show mpls l2transport vc** **Example:** `Router# show mpls l2transport vc`	Displays information about Ethernet over MPLS port mode.

Examples

In the following example, the output of the **show mpls l2transport vc detail**command is displayed:

```
Router# show mpls l2transport vc detail
Local interface: Gi4/0.1 up, line protocol up, Eth VLAN 2 up
Destination address: 10.1.1.1, VC ID: 2, VC status: up
.
.
.
Local interface: Gi8/0/1 up, line protocol up, Ethernet up
  Destination address: 10.1.1.1, VC ID: 8, VC status: up
```

Configuring Ethernet over MPLS with VLAN ID Rewrite

The VLAN ID rewrite feature enables you to use VLAN interfaces with different VLAN IDs at both ends of the tunnel.

The Cisco 12000 series router requires you to configure VLAN ID rewrite manually, as described in the following sections.

The following routers automatically perform VLAN ID rewrite on the disposition PE router. No configuration is required:

- Cisco 7200 series routers.
- Cisco 7500 series routers.
- Cisco 10720 series routers.
- Routers supported on Cisco IOS Release 12.4(11)T. (Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support.)

The following sections explain how to configure the VLAN ID rewrite feature:

Configuring Ethernet over MPLS with VLAN ID Rewrite for Cisco 12k Routers for 12.0(29)S and Earlier Releases

Use the following guidelines for the VLAN ID rewrite feature for the Cisco 12000 series routers in Cisco IOS releases earlier than 12.0(29)S:

- The IP Service Engine (ISE) 4-port Gigabit Ethernet line card performs the VLAN ID rewrite on the disposition side at the edge-facing line card.
- The engine 2 3-port Gigabit Ethernet line card performs the VLAN ID rewrite on the imposition side at the edge-facing line card.

The VLAN ID rewrite functionality requires that both ends of the Ethernet over MPLS connections be provisioned with the same line cards. Make sure that both edge-facing ends of the virtual circuit use either the engine 2 or ISE Ethernet line card. The following example shows the system flow with the VLAN ID rewrite feature:

- The ISE 4-port Gigabit Ethernet line card:

Traffic flows from VLAN1 on CE1 to VLAN2 on CE2. As the frame reaches the edge-facing line card of the disposition router PE2, the VLAN ID in the dot1Q header changes to the VLAN ID assigned to VLAN2.

- The engine 2 3-port Gigabit Ethernet line card:

Traffic flows from VLAN1 on CE1 to VLAN2 on CE2. As the frame reaches the edge-facing line card of the imposition router PE1, the VLAN ID in the dot1Q header changes to the VLAN ID assigned to VLAN2.

For the Cisco 12000 series router engine 2 3-port Gigabit Ethernet line card, you must issue the **remote circuit id** command as part of the Ethernet over MPLS VLAN ID rewrite configuration.

Configuring Ethernet over MPLS with VLAN ID Rewrite for Cisco 12k Routers for 12.0(30)S and Later Releases

In Cisco IOS Release 12.0(30)S, the following changes to VLAN ID rewrite were implemented:

- The ISE 4-port Gigabit Ethernet line card can perform VLAN ID rewrite at both the imposition and disposition sides of the edge-facing router.
- The **remote circuit id** command is not required as part of the Ethernet over MPLS VLAN ID rewrite configuration, as long as both PE routers are running Cisco IOS Release 12.0(30)S. The VLAN ID rewrite feature is implemented automatically when you configure Ethernet over MPLS.
- The VLAN ID rewrite feature in Cisco IOS Release 12.0(30)S can interoperate with routers that are running earlier releases. If you have a PE router at one end of the circuit that is using an earlier Cisco IOS release and the **remote circuit id**command, the other PE can run Cisco IOS Release 12.0(30)S and still perform VLAN ID rewrite.
- You can mix the line cards on the PE routers, as shown in the following table

Table 6: Supported Line Cards for VLAN ID Rewrite Feature:

If PE1 Has These Line Cards	Then PE2 Can Use These Line Cards
Engine 2 3-port Gigabit Ethernet line card or ISE 4-port Gigabit Ethernet line card	Engine 2 3-port Gigabit Ethernet line card or ISE 4-port Gigabit Ethernet line card
ISE 4-port Gigabit Ethernet line card	Any Cisco 12000 series router line card

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot /interface.subinterface*
4. **encapsulation dot1q** *vlan-id*
5. **xconnect** *peer-router-id vcid* **encapsulation mpls**
6. **remote circuit id** *remote-vlan-id*
7. **exit**
8. **exit**
9. **exit**
10. **show controllers eompls forwarding-table**

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 gigabitethernet** *slot /interface.subinterface* **Example:** `Router(config)# interface gigabitethernet4/0.1`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. • Make sure the subinterfaces between the CE and PE routers that are running Ethernet over MPLS are in the same subnet. All other subinterfaces and backbone routers do not need to be in the same subnet.
Step 4	**encapsulation dot1q** *vlan-id*	Enables the subinterface to accept 802.1Q VLAN packets.

	Command or Action	Purpose
	Example: `Router(config-subif)# encapsulation dot1q 100`	• Make sure the subinterface on the adjoining CE router is on the same VLAN as this PE router.
Step 5	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-subif)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC and enters xconnect configuration mode. • The syntax for this command is the same as for all other Layer 2 transports.
Step 6	**remote circuit id** *remote-vlan-id* **Example:** `Router(config-subif-xconn)# remote circuit id 101`	Enables you to use VLAN interfaces with different VLAN IDs at both ends of the tunnel. • This command is required only for the Cisco 12000 series router engine 2 3-port Gigabit Ethernet line card.
Step 7	**exit** **Example:** `Router(config-subif-xconn)# exit`	Exits xconnect configuration mode.
Step 8	**exit** **Example:** `Router(config-subif)# exit`	Exits subinterface configuration mode.
Step 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 10	**show controllers eompls forwarding-table** **Example:** `Router# execute slot 0 show controllers eompls forwarding-table`	Displays information about VLAN ID rewrite.

Examples

The command output of the **show controllers eompls forwarding-table**command in the following example shows VLAN ID rewrite configured on the Cisco 12000 series routers with an engine 2 3-port Gigabit Ethernet line card. In the following example, the bolded command output show the VLAN ID rewrite information.

On PE1

```
Router# execute slot 0 show controllers eompls forwarding-table 0 2
Port # 0, VLAN-ID # 2, Table-index 2
EoMPLS configured: 1
tag_rew_ptr             = D001BB58
Leaf entry?     = 1
FCR index       = 20
           **tagrew_psa_addr    = 0006ED60
           **tagrew_vir_addr    = 7006ED60
           **tagrew_phy_addr    = F006ED60
        [0-7] loq 8800 mtu 4458  oq 4000 ai 3 oi 04019110 (encaps size 4)
        cw-size 4 vlanid-rew 3
        gather A30 (bufhdr size 32 EoMPLS (Control Word) Imposition profile 81)
        2 tag: 18 18
        counters 1182, 10 reported 1182, 10.
    Local OutputQ (Unicast):    Slot:2  Port:0  RED queue:0  COS queue:0
    Output Q (Unicast):         Port:0          RED queue:0  COS queue:0
```

On PE2

```
Router# execute slot 0 show controllers eompls forwarding-table 0 3

Port # 0, VLAN-ID # 3, Table-index 3
EoMPLS configured: 1
tag_rew_ptr             = D0027B90
Leaf entry?     = 1
FCR index       = 20
           **tagrew_psa_addr    = 0009EE40
           **tagrew_vir_addr    = 7009EE40
           **tagrew_phy_addr    = F009EE40
        [0-7] loq 9400 mtu 4458  oq 4000 ai 8 oi 84000002 (encaps size 4)
        cw-size 4 vlanid-rew 2
        gather A30 (bufhdr size 32 EoMPLS (Control Word) Imposition profile 81)
        2 tag: 17 18
        counters 1182, 10 reported 1182, 10.
    Local OutputQ (Unicast):    Slot:5  Port:0  RED queue:0  COS queue:0
    Output Q (Unicast):         Port:0          RED queue:0  COS queue:0
```

Configuring per-Subinterface MTU for Ethernet over MPLS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *subslot* / *port* [**.** *subinterface*]
4. **mtu** *mtu-value*
5. **interface gigabitethernet** *slot* / *subslot* / *port* [**.** *subinterface*]
6. **encapsulation dot1q** *vlan-id*
7. **xconnect** *peer-router-id vcid* **encapsulation mpls**
8. **mtu** *mtu-value*
9. **end**
10. **show mpls l2transport binding**

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 gigabitethernet** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface gigabitethernet4/0/0`	Specifies the Gigabit Ethernet interface and enters interface configuration mode.
Step 4	**mtu** *mtu-value* **Example:** `Router(config-if)# mtu 2000`	Specifies the MTU value for the interface. The MTU value specified at the interface level can be inherited by a subinterface.
Step 5	**interface gigabitethernet** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config-if)# interface gigabitethernet4/0/0.1`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. Make sure the subinterface on the adjoining CE router is on the same VLAN as this PE router.
Step 6	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-subif)# encapsulation dot1q 100`	Enables the subinterface to accept 802.1Q VLAN packets. The subinterfaces between the CE and PE routers that are running Ethernet over MPLS must be in the same subnet. All other subinterfaces and backbone routers need not be.
Step 7	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-subif)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports. Enters xconnect subinterface configuration mode.

	Command or Action	Purpose
Step 8	**mtu** *mtu-value* **Example:** `Router(config-if-xconn)# mtu 1400`	Specifies the MTU for the VC.
Step 9	**end** **Example:** `Router(config-if-xconn)# end`	Exits to privileged EXEC mode.
Step 10	**show mpls l2transport binding** **Example:** `Router# show mpls l2transport binding`	Displays the MTU values assigned to the local and remote interfaces.

Configuring Frame Relay over MPLS with DLCI-to-DLCI Connections

Frame Relay over MPLS encapsulates Frame Relay PDUs in MPLS packets and forwards them across the MPLS network. For Frame Relay, you can set up data-link connection identifier (DLCI)-to-DLCI connections or port-to-port connections. With DLCI-to-DLCI connections, the PE routers manipulate the packet by removing headers, adding labels, and copying control word elements from the header to the PDU.

Perform this task to configure Frame Relay over MPLS with DLCI-to-DLCI connections.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **frame-relay switching**
4. **interface serial** *slot /port*
5. **encapsulation frame-relay** [**cisco** | **ietf**]
6. **frame-relay intf-type dce**
7. **exit**
8. **connect** *connection-name interface dlci* **l2transport**
9. **xconnect** *peer-router-id vcid* **encapsulation mpls**
10. **end**

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	**frame-relay switching** **Example:** `Router(config)# frame-relay switching`	Enables PVC switching on a Frame Relay device.
Step 4	**interface serial** *slot /port* **Example:** `Router(config)# interface serial3/1`	Specifies a serial interface and enters interface configuration mode.
Step 5	**encapsulation frame-relay** [**cisco** | **ietf**] **Example:** `Router(config-if)# encapsulation frame-relay ietf`	Specifies Frame Relay encapsulation for the interface. • You can specify different types of encapsulations. You can set one interface to Cisco encapsulation and the other interface to IETF encapsulation.
Step 6	**frame-relay intf-type dce** **Example:** `Router(config-if)# frame-relay intf-type dce`	Specifies that the interface is a DCE switch. • You can also specify the interface to support Network-to-Network Interface (NNI) and DTE connections.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits from interface configuration mode.
Step 8	**connect** *connection-name interface dlci* **l2transport** **Example:** `Router(config)# connect fr1 serial5/0 1000 l2transport`	Defines connections between Frame Relay PVCs and enters connect configuration mode. • Using the **l2transport** keyword specifies that the PVC will not be a locally switched PVC, but will be tunneled over the backbone network.

	Command or Action	Purpose
		• The *connection-name* argument is a text string that you provide. • The *interface* argument is the interface on which a PVC connection will be defined. • The *dlci*argument is the DLCI number of the PVC that will be connected.
Step 9	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-fr-pw-switching)# xconnect 10.0.0.1 123 encapsulation mpls`	Creates the VC to transport the Layer 2 packets. • In a DLCI-to DLCI connection type, Frame Relay over MPLS uses the **xconnect** command in connect configuration mode.
Step 10	**end** **Example:** `Router(config-fr-pw-switching)# end`	Exits connect configuration mode and returns to privileged EXEC mode.

Configuring Frame Relay over MPLS with Port-to-Port Connections

Frame Relay over MPLS encapsulates Frame Relay PDUs in MPLS packets and forwards them across the MPLS network. For Frame Relay, you can set up DLCI-to-DLCI connections or port-to-port connections. With port-to-port connections, you use HDLC mode to transport the Frame Relay encapsulated packets. In HDLC mode, the whole HDLC packet is transported. Only the HDLC flags and FCS bits are removed. The contents of the packet are not used or changed, including the backward explicit congestion notification (BECN), forward explicit congestion notification (FECN) and discard eligibility (DE) bits.

Perform this task to set up Frame Relay port-to-port connections.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *slot /port*
4. **encapsulation hdlc**
5. **xconnect** *peer-router-id vcid* **encapsulation mpls**
6. **end**

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 serial** *slot /port* **Example:** `Router(config)# interface serial5/0`	Specifies a serial interface and enters interface configuration mode.
Step 4	**encapsulation hdlc** **Example:** `Router(config-if)# encapsulation hdlc`	Specifies that Frame Relay PDUs will be encapsulated in HDLC packets.
Step 5	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if)# xconnect 10.0.0.1 123 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring HDLC and PPP over MPLS

With HDLC over MPLS, the whole HDLC packet is transported. The ingress PE router removes only the HDLC flags and FCS bits. The contents of the packet are not used or changed.

With PPP over MPLS, the ingress PE router removes the flags, address, control field, and the FCS.

Note The following restrictions pertain to the HDLC over MPLS feature:

- Asynchronous interfaces are not supported.
- You must configure HDLC over MPLS on router interfaces only. You cannot configure HDLC over MPLS on subinterfaces.

The following restrictions pertain to the PPP over MPLS feature:

- Zero hops on one router is not supported. However, you can have back-to-back PE routers.
- Asynchronous interfaces are not supported. The connections between the CE and PE routers on both ends of the backbone must have similar link layer characteristics. The connections between the CE and PE routers must both be synchronous.
- Multilink PPP (MLP) is not supported.
- You must configure PPP on router interfaces only. You cannot configure PPP on subinterfaces.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *slot /port*
4. Do one of the following:
   - **encapsulation ppp**
   -
   - **encapsulation hdlc**
5. **xconnect** *peer-router-id vcid* **encapsulation mpls**
6. **end**

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	**interface serial** *slot /port* **Example:** `Router(config)# interface serial5/0`	Specifies a serial interface and enters interface configuration mode. • You must configure HDLC and PPP over MPLS on router interfaces only. You cannot configure HDLC over MPLS on subinterfaces.

Step 4	Do one of the following:
	• **encapsulation ppp**
	•
	• **encapsulation hdlc**
	
	Example:
	`Router(config-if)# encapsulation ppp`
	
	Example:
	`or`
	
	Example:
	
	Example:
	`Router(config-if)# encapsulation hdlc`
		Specifies HDLC or PPP encapsulation and enters connect configuration mode.

Step 5	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-fr-pw-switching)# xconnect 10.0.0.1 123 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.
Step 6	**end**	Exits connect configuration mode and returns to privileged EXEC mode.

Configuring Tunnel Selection

The tunnel selection feature allows you to specify the path that traffic uses. You can specify either an MPLS TE tunnel or destination IP address or domain name server (DNS) name.

You also have the option of specifying whether the VCs should use the default path (the path LDP uses for signaling) if the preferred path is unreachable. This option is enabled by default; you must explicitly disable it.

You configure tunnel selection when you set up the pseudowire class. You enable tunnel selection with the **preferred-path** command. Then, you apply the pseudowire class to an interface that has been configured to transport AToM packets.

The following guidelines provide more information about configuring tunnel selection:

- The **preferred-path** command is available only if the pseudowire encapsulation type is MPLS.
- This tunnel selection feature is enabled when you exit from pseudowire mode.
- The selected path should be an LSP destined to the peer PE router.
- The selected tunnel must be an MPLS TE tunnel.
- If you select a tunnel, the tunnel tailend must be on the remote PE router.
- If you specify an IP address, that address must be the IP address of the loopback interface on the remote PE router. The address must have a /32 mask. There must be an LSP destined to that selected address. The LSP need not be a TE tunnel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *name*
4. **encapsulation mpls**
5. **preferred-path** {**interface tunnel** *tunnel-number* | **peer** {*ip-address* | *host-name*}} [**disable-fallback**]
6. **exit**
7. **interface** *slot /port*
8. **encapsulation** *encapsulation-type*
9. **xconnect** *peer-router-id vcid* **pw-class name**
10. **end**

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	**pseudowire-class** *name* **Example:** `Router(config)# pseudowire-class ts1`	Establishes a pseudowire class with a name that you specify and enters pseudowire configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation. • For AToM, the encapsulation type is **mpls**.
Step 5	**preferred-path** {**interface tunnel** *tunnel-number* | **peer** {*ip-address* | *host-name*}} [**disable-fallback**] **Example:** `Router(config-pw-class)# preferred path peer 10.18.18.18`	Specifies the MPLS traffic engineering tunnel or IP address or hostname to be used as the preferred path.
Step 6	**exit** **Example:** `Router(config-pw-class)# exit`	Exits from pseudowire configuration mode.
Step 7	**interface** *slot /port* **Example:** `Router(config)# interface atm1/1`	Specifies an interface and enters interface configuration mode.
Step 8	**encapsulation** *encapsulation-type* **Example:** `Router(config-if)# encapsulation aal5`	Specifies the encapsulation for the interface.
Step 9	**xconnect** *peer-router-id vcid* **pw-class name** **Example:** `Router(config-if)# xconnect 10.0.0.1 123 pw-class ts1`	Binds the attachment circuit to a pseudowire VC.
Step 10	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to Privileged EXEC mode.

Examples

In the following example, the **show mpls l2transport vc** command shows the following information about the VCs:

- VC 101 has been assigned a preferred path called Tunnel1. The default path is disabled, because the preferred path specified that the default path should not be used if the preferred path fails.
- VC 150 has been assigned an IP address of a loopback address on PE2. The default path can be used if the preferred path fails.

In the following example, command output that is bolded shows the preferred path information.

```
Router# show mpls l2transport vc detail
Local interface: Gi0/0/0.1 up, line protocol up, Eth VLAN 222 up
  Destination address: 10.16.16.16, VC ID: 101, VC status: up
    Preferred path: Tunnel1,  active
    Default path: disabled
    Tunnel label: 3, next hop point2point
    Output interface: Tu1, imposed label stack {17 16}
  Create time: 00:27:31, last status change time: 00:27:31
  Signaling protocol: LDP, peer 10.16.16.16:0 up
    MPLS VC labels: local 25, remote 16
    Group ID: local 0, remote 6
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 10, send 10
    byte totals:   receive 1260, send 1300
    packet drops:  receive 0, send 0
Local interface: AT1/0/0 up, line protocol up, ATM AAL5 0/50 up
  Destination address: 10.16.16.16, VC ID: 150, VC status: up
    Preferred path: 10.18.18.18, active
    Default path: ready
    Tunnel label: 3, next hop point2point
    Output interface: Tu2, imposed label stack {18 24}
  Create time: 00:15:08, last status change time: 00:07:37
  Signaling protocol: LDP, peer 10.16.16.16:0 up
    MPLS VC labels: local 26, remote 24
    Group ID: local 2, remote 0
    MTU: local 4470, remote 4470
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 0, send 0
    byte totals:   receive 0, send 0
    packet drops:  receive 0, send 0
```

Troubleshooting Tips

You can use the **debug mpls l2transport vc event**command to troubleshoot tunnel selection. For example, if the tunnel interface that is used for the preferred path is shut down, the default path is enabled. The **debug mpls l2transport vc event**command provides the following output:

```
AToM SMGR [10.2.2.2, 101]: Processing imposition update, vc_handle 62091860, update_action
 3, remote_vc_label 16
AToM SMGR [10.2.2.2, 101]: selected route no parent rewrite: tunnel not up
AToM SMGR [10.2.2.2, 101]: Imposition Programmed, Output Interface: Et3/2
```

Setting Experimental Bits with AToM

MPLS AToM uses the three experimental bits in a label to determine the queue of packets. You statically set the experimental bits in both the VC label and the LSP tunnel label, because the LSP tunnel label might be removed at the penultimate router. The following sections explain the transport-specific implementations of the EXP bits.

Note For information about setting EXP bits on the Cisco 12000 series router for Cisco IOS Release 12.0(30)S, see the AToM: L2 QoS feature module.

Note The following restrictions apply to ATM AAL5 over MPLS with EXP bits:

- ATM AAL5 over MPLS allows you to statically set the experimental bits.
- If you do not assign values to the experimental bits, the priority bits in the header's "tag control information" field are set to zero.
- On the Cisco 7500 series routers, distributed Cisco Express Forwarding must be enabled before you set the experimental bits.

The following restrictions apply to ATM Cell Relay over MPLS with EXP bits:

- ATM Cell Relay over MPLS allows you to statically set the experimental bits in VC, PVP, and port modes.
- If you do not assign values to the experimental bits, the priority bits in the header's "tag control information" field are set to zero.
- On the Cisco 7500 series routers, distributed Cisco Express Forwarding must be enabled before you set the experimental bits.

The following restrictions apply to Ethernet over MPLS with EXP bits:

On the Cisco 7200 and 7500 Series Routers

- Ethernet over MPLS allows you to set the EXP bits by using either of the following methods:
  - Writing the priority bits into the experimental bit field, which is the default.
  - Using the **match any**command with the **set mpls exp** command.
- If you do not assign values to the experimental bits, the priority bits in the 802.1Q header's "tag control information" field are written into the experimental bit fields.
- On the Cisco 7500 series routers, distributed Cisco Express Forwarding must be enabled before you set the experimental bits.

On the Cisco 10720 Internet Router

The table below lists the commands that are supported on the Cisco 10720 Internet router for Ethernet over MPLS. The letter Y means that the command is supported on that interface. A dash (--) means that command is not supported on that interface.

Note The **match cos**command is supported only on subinterfaces, not main interfaces.

Table 7: Commands Supported on the Cisco 10720 Router for Ethernet over MPLS

Commands	Imposition	Disposition		
Traffic Matching Commands	In	Out	In	Out

Commands	Imposition	Disposition		
match any	Y	Y	Y	Y
match cos	Y	--	--	--
match input-interface	--	--	Y	Y
match mpls exp	--	Y	Y	--
match qos-group	--	Y	--	Y
Traffic Action Commands	In	Out	In	Out
set cos	--	--	--	Y
set mpls exp	Y	--	--	--
set qos-group	Y	--	Y	--
set srp-priority	--	Y	--	--

The following restrictions apply to Frame Relay over MPLS and EXP bits:

- If you do not assign values to the experimental bits, the priority bits in the header's “tag control information” field are set to zero.
- On the Cisco 7500 series routers, distributed Cisco Express Forwarding must be enabled before you set the experimental bits.

The following restrictions apply to HDLC over MPLS and PPP over MPLS and EXP bits:

- If you do not assign values to the experimental bits, zeros are written into the experimental bit fields.
- On the Cisco 7500 series routers, enable distributed Cisco Express Forwarding before setting the experimental bits.

Set the experimental bits in both the VC label and the LSP tunnel label. You set the experimental bits in the VC label, because the LSP tunnel label might be removed at the penultimate router. Perform this task to set the experimental bits.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-name*
4. **match any**
5. **exit**
6. **policy-map** *policy-name*
7. **class** *class-name*
8. **set mpls experimental** *value*
9. **exit**
10. **exit**
11. **interface** *slot /port*
12. **service-policy input** *policy-name*
13. **exit**
14. **exit**
15. **show policy-map interface** *interface-name* [**vc** [*vpi*/] *vci*] [**dlci** *dlci*] [**input** | **output**]

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	**class-map** *class-name* **Example:** `Router(config)# class-map class1`	Specifies the user-defined name of the traffic class and enters class map configuration mode.
Step 4	**match any** **Example:** `Router(config-cmap)# match any`	Specifies that all packets will be matched. • Use only the **any** keyword. Other keywords might cause unexpected results.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Exits class map configuration mode.
Step 6	**policy-map** *policy-name* **Example:** `Router(config)# policy-map policy1`	Specifies the name of the traffic policy to configure and enters policy-map configuration mode.
Step 7	**class** *class-name* **Example:** `Router(config-pmap)# class class1`	Specifies the name of the predefined traffic that was configured with the **class-map** command and was used to classify traffic to the traffic policy specified, and enters policy-map class configuration mode.
Step 8	**set mpls experimental** *value* **Example:** `Router(config-pmap-c)# set mpls experimental 7`	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
Step 9	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 10	**exit** **Example:** `Router(config-pmap)# exit`	Exits policy-map configuration mode.
Step 11	**interface** *slot /port* **Example:** `Router(config)# interface atm4/0`	Specifies the interface and enters interface configuration mode.
Step 12	**service-policy input** *policy-name* **Example:** `Router(config-if)# service-policy input policy1`	Attaches a traffic policy to an interface.

	Command or Action	Purpose
Step 13	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 14	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 15	**show policy-map interface** *interface-name* [**vc** [*vpi*/] *vci*] [**dlci** *dlci*] [**input** | **output**] **Example:** `Router# show policy-map interface serial3/0`	Displays the traffic policy attached to an interface.

Setting the Frame Relay Discard Eligibility Bit on the Cisco 7200 and 7500 Series Routers

You can use the DE bit in the address field of a Frame Relay frame to prioritize frames in congested Frame Relay networks. The Frame Relay DE bit has only one bit and can therefore only have two settings, 0 or 1. If congestion occurs in a Frame Relay network, frames with the DE bit set to 1 are discarded before frames with the DE bit set to 0. Therefore, important traffic should have the DE bit set to 0, and less important traffic should be forwarded with the DE bit set at 1. The default DE bit setting is 0. You can change the DE bit setting to 1 with the **set fr-de** command.

Note The **set fr-de** command can be used only in an output service policy.

Perform this task to set the Frame Relay DE bit on the Cisco 7200 and 7500 series routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-name*
4. **class** *class-name*
5. **set fr-de**
6. **end**

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	**policy-map** *policy-name* **Example:** `Router(config)# policy-map policy1`	Specifies the name of the traffic policy to configure and enters policy-map configuration mode. • Names can be a maximum of 40 alphanumeric characters.
Step 4	**class** *class-name* **Example:** `Router(config-pmap)# class class1`	Specifies the name of a predefined traffic class and enters policy-map class configuration mode.
Step 5	**set fr-de** **Example:** `Router(config-pmap-c)# set fr-de`	Sets the Frame Relay DE bit setting for all packets that match the specified traffic class from 0 to 1.
Step 6	**end** **Example:** `Router(config-pmap-c)# end`	Exits policy-map class configuration mode and returns to privileged EXEC mode.

Matching the Frame Relay DE Bit on the Cisco 7200 and 7500 Series Routers

You can use the **match fr-de**command to enable frames with a DE bit setting of 1 to be considered a member of a defined class and forwarded according to the specifications set in the service policy.

Perform this task to match frames with the FR DE bit set to 1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-map-name*
4. **match fr-de**
5. **end**

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	**class-map** *class-map-name* **Example:** `Router(config)# class-map de-bits`	Specifies the name of a predefined traffic class and enters class-map configuration mode.
Step 4	**match fr-de** **Example:** `Router(config-cmap)# match fr-de`	Classifies all frames with the DE bit set to 1.
Step 5	**end** **Example:** `Router(config-cmap)# end`	Exits class-map configuration mode and returns to privileged EXEC mode.

Enabling the Control Word

You can enable the control word for dynamic and static pseudowires under a pseudowire class. Use the **control-word** command to enable, disable, or set a control word to autosense mode. If you do not enable a control word, autosense is the default mode for the control word.

Perform this task to enable a control word.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class cw_enable**
4. **encapsulation mpls**
5. **control-word**
6. **exit**
7. **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	**pseudowire-class cw_enable** **Example:** `Router(config)# pseudowire-class cw_enable`	Enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation. • For AToM, the encapsulation type is mpls.
Step 5	**control-word** **Example:** `Router(config-pw-class)# control-word`	Enables the control word.
Step 6	**exit** **Example:** `Router(config-pw-class)# exit`	Exits pseudowire class configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Configuration Examples for Any Transport over MPLS

Example ATM AAL5 over MPLS

ATM AAL5 over MPLS on PVCs

The following example shows how to enable ATM AAL5 over MPLS on an ATM PVC:

```
enable
 configure terminal
 interface atm1/
0
 pvc 1/
200 l2transport
 encapsulation aal5
 xconnect 10.13.13.13 100 encapsulation mpls
```

ATM AAL5 over MPLS in VC Class Configuration Mode

The following example shows how to configure ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to an interface.

```
enable
configure terminal
vc-class atm aal5class
encapsulation aal5
interface atm1/
0
class-int aal5class
pvc 1/
200 l2transport
xconnect 10.13.13.13 100 encapsulation mpls
```

The following example shows how to configure ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to a PVC.

```
enable
configure terminal
vc-class atm aal5class
encapsulation aal5
interface atm1/
0
pvc 1/
200 l2transport
class-vc aal5class
xconnect 10.13.13.13 100 encapsulation mpls
```

Example OAM Cell Emulation for ATM AAL5 over MPLS

OAM Cell Emulation for ATM AAL5 over MPLS on PVCs

The following example shows how to enable OAM cell emulation on an ATM PVC:

```
interface ATM 1/0/0
pvc 1/200 l2transport
encapsulation aal5
xconnect 10.13.13.13 100 encapsulation mpls
oam-ac emulation-enable
oam-pvc manage
```

The following example shows how to set the rate at which an AIS cell is sent every 30 seconds:

```
interface ATM 1/0/0
pvc 1/200 l2transport
encapsulation aal5
xconnect 10.13.13.13 100 encapsulation mpls
oam-ac emulation-enable 30
oam-pvc manage
```

OAM Cell Emulation for ATM AAL5 over MPLS in VC Class Configuration Mode

The following example shows how to configure OAM cell emulation for ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to an interface.

```
enable
configure terminal
vc-class atm oamclass
encapsulation aal5
oam-ac emulation-enable 30
oam-pvc manage
interface atm1/0
class-int oamclass
pvc 1/200 l2transport
xconnect 10.13.13.13 100 encapsulation mpls
```

The following example shows how to configure OAM cell emulation for ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to a PVC.

```
enable
configure terminal
vc-class atm oamclass
encapsulation aal5
oam-ac emulation-enable 30
oam-pvc manage
interface atm1/0
pvc 1/200 l2transport
class-vc oamclass
xconnect 10.13.13.13 100 encapsulation mpls
```

The following example shows how to configure OAM cell emulation for ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to an interface. One PVC is configured with OAM cell emulation at an AIS rate of 10. That PVC uses the AIS rate of 10 instead of 30.

```
enable
configure terminal
vc-class atm oamclass
encapsulation aal5
oam-ac emulation-enable 30
oam-pvc manage
interface atm1/0
class-int oamclass
pvc 1/200 l2transport
```

```
oam-ac emulation-enable 10
xconnect 10.13.13.13 100 encapsulation mpls
```

Example ATM Cell Relay over MPLS

ATM Cell Relay over MPLS in VC Mode Using VC Class Configuration Mode

The following example shows how to configure ATM cell relay over MPLS in VC class configuration mode. The VC class is then applied to an interface.

```
enable
configure terminal
vc-class atm cellrelay
encapsulation aal0
interface atm1/0
class-int cellrelay
pvc 1/200 l2transport
xconnect 10.13.13.13 100 encapsulation mpls
```

The following example shows how to configure ATM cell relay over MPLS in VC class configuration mode. The VC class is then applied to a PVC.

```
enable
configure terminal
vc-class atm cellrelay
encapsulation aal0
interface atm1/0
pvc 1/200 l2transport
class-vc cellrelay
xconnect 10.13.13.13 100 encapsulation mpls
```

ATM Cell Relay over MPLS in PVP Mode

The following example shows how to transport single ATM cells over a virtual path:

```
pseudowire-class vp-cell-relay
encapsulation mpls
interface atm 5/0
atm pvp 1 l2transport
xconnect 10.0.0.1 123 pw-class vp-cell-relay
```

ATM Cell Relay over MPLS in Port Mode

The following example shows how to configure interface ATM 5/0 to transport ATM cell relay packets:

```
pseudowire-class atm-cell-relay
encapsulation mpls
interface atm 5/0
xconnect 10.0.0.1 123 pw-class atm-cell-relay
```

The following example shows how to configure interface ATM 9/0/0 to transport ATM cell relay packets on a Cisco 7600 series router, where you must specify the interface ATM slot, bay, and port:

```
pseudowire-class atm-cell-relay
encapsulation mpls
interface atm 9/0/0
xconnect 10.0.0.1 500 pw-class atm-cell-relay
```

Example ATM Single Cell Relay over MPLS

ATM Packed Cell Relay over MPLS in VC Mode

The following example shows that ATM PVC 1/100 is an AToM cell relay PVC. There are three timers set up, with values of 1000 milliseconds, 800 milliseconds, and 500 milliseconds, respectively. The **cell-packing** command specifies that five ATM cells are to be packed into an MPLS packet. The **cell-packing** command also specifies that timer 1 is to be used.

```
interface atm 1/0
shutdown
atm mcpt-timer 1000 800 500
no shutdown
pvc 1/100 l2transport
encapsulation aal0
xconnect 10.0.0.1 123 encapsulation mpls
cell-packing 5 mcpt-timer 1
```

ATM Packed Cell Relay over MPLS in VC Mode Using VC Class Configuration Mode

The following example shows how to configure ATM cell relay over MPLS with cell packing in VC class configuration mode. The VC class is then applied to an interface.

```
enable
configure terminal
vc-class atm cellpacking
encapsulation aal0
cell-packing 10 mcpt-timer 1
interface atm1/0
shutdown
atm mcpt-timers 100 200 250
no shutdown
class-int cellpacking
pvc 1/200 l2transport
xconnect 10.13.13.13 100 encapsulation mpls
```

The following example shows how to configure ATM cell relay over MPLS in VC class configuration mode. The VC class is then applied to a PVC.

```
enable
configure terminal
vc-class atm cellpacking
encapsulation aal0
cell-packing 10 mcpt-timer 1
interface atm1/0
shutdown
atm mcpt-timers 100 200 250
no shutdown
pvc 1/200 l2transport
class-vc cellpacking
xconnect 10.13.13.13 100 encapsulation mpls
```

ATM Packed Cell Relay over MPLS in VP Mode

The following example shows packed cell relay enabled on an interface configured for PVP mode. The **cell-packing** command specifies that 10 ATM cells are to be packed into an MPLS packet. The **cell-packing** command also specifies that timer 2 is to be used.

```
interface atm 1/0
shutdown
atm mcpt-timer 1000 800 500
```

```
no shutdown
atm pvp 100 l2transport
xconnect 10.0.0.1 234 encapsulation mpls
cell-packing 10 mcpt-timer 2
```

ATM Packed Cell Relay over MPLS in Port Mode

The following example shows packed cell relay enabled on an interface set up for port mode. The **cell-packing** command specifies that 10 ATM cells are to be packed into an MPLS packet. The **cell-packing** command also specifies that timer 2 is to be used.

```
interface atm 5/0
shutdown
atm mcpt-timer 1000 800 500
no shutdown
cell-packing 10 mcpt-timer 2
xconnect 10.0.0.1 123 encapsulation mpls
```

Example Ethernet over MPLS

Ethernet over MPLS in Port Mode

The following example shows how to configure VC 123 in Ethernet port mode:

```
pseudowire-class ethernet-port
encapsulation mpls

int gigabitethernet1/0
xconnect 10.0.0.1 123 pw-class ethernet-port
```

Ethernet over MPLS with VLAN ID Rewrite

The following example shows how to configure VLAN ID rewrite on peer PE routers with Cisco 12000 series router engine 2 3-port Gigabit Ethernet line cards.

PE1	PE2

`interface GigabitEthernet0/0.2`
`encapsulation dot1Q 2`
`no ip directed-broadcast`
`no cdp enable`
`xconnect 10.5.5.5 2 encapsulation mpls`
`remote circuit id 3`
	`interface GigabitEthernet3/0.2`
	`encapsulation dot1Q 3`
	`no ip directed-broadcast`
	`no cdp enable`
	`xconnect 10.3.3.3 2 encapsulation mpls`
	`remote circuit id 2`

Example Tunnel Selection

The following example shows how to set up two preferred paths for PE1. One preferred path specifies an MPLS traffic engineering tunnel. The other preferred path specifies an IP address of a loopback address on PE2. There is a static route configured on PE1 that uses a TE tunnel to reach the IP address on PE2.

PE1 Configuration

```
mpls label protocol ldp
mpls traffic-eng tunnels
```

```
tag-switching tdp router-id Loopback0
pseudowire-class pw1
 encapsulation mpls
 preferred-path interface Tunnel1 disable-fallback
!
pseudowire-class pw2
 encapsulation mpls
 preferred-path peer 10.18.18.18
!
interface Loopback0
 ip address 10.2.2.2 255.255.255.255
 no ip directed-broadcast
 no ip mroute-cache
!
interface Tunnel1
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 10.16.16.16
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 1500
 tunnel mpls traffic-eng path-option 1 explicit name path-tu1
!
interface Tunnel2
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 10.16.16.16
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 1500
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface gigabitethernet0/0/0
 no ip address
 no ip directed-broadcast
 no negotiation auto
!
interface gigabitethernet0/0/0.1
 encapsulation dot1Q 222
 no ip directed-broadcast
 xconnect 10.16.16.16 101 pw-class pw1
!
interface ATM1/0/0
 no ip address
 no ip directed-broadcast
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
 pvc 0/50 l2transport
  encapsulation aal5
  xconnect 10.16.16.16 150 pw-class pw2
!
interface Ethernet2/0/1
 ip address 10.0.0.1 255.255.255.0
 no ip directed-broadcast
 tag-switching ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 15000 15000
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.0.0.255 area 0
 network 10.2.2.2 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
!
ip route 10.18.18.18 255.255.255.255 Tunnel2
!
ip explicit-path name path-tu1 enable
 next-address 10.0.0.1
 index 3 next-address 10.0.0.1
```

PE2 Configuration

```
mpls label protocol ldp
mpls traffic-eng tunnels
mpls ldp router-id Loopback0
interface Loopback0
 ip address 10.16.16.16 255.255.255.255
 no ip directed-broadcast
 no ip mroute-cache
!
interface Loopback2
 ip address 10.18.18.18 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet3/1
 ip address 10.0.0.2 255.255.255.0
 no ip directed-broadcast
 mpls traffic-eng tunnels
 mpls ip
 no cdp enable
 ip rsvp bandwidth 15000 15000
!
interface Ethernet3/3
 no ip address
 no ip directed-broadcast
 no cdp enable
!
interface Ethernet3/3.1
 encapsulation dot1Q 222
 no ip directed-broadcast
 no cdp enable
 mpls l2transport route 10.2.2.2 101
!
interface ATM5/0
 no ip address
 no ip directed-broadcast
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
 pvc 0/50 l2transport
  encapsulation aal5
  xconnect 10.2.2.2 150 encapsulation mpls
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.0.0.255 area 0
 network 10.16.16.16 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
```

Example Setting Frame Relay Discard Eligibility Bit on the Cisco 7200 and 7500 Series Routers

The following example shows how to configure the service policy called set-de and attach it to an interface. In this example, the class map called data evaluates all packets exiting the interface for an IP precedence value of 1. If the exiting packet has been marked with the IP precedence value of 1, the packet's DE bit is set to 1.

```
class-map data
match ip precedence 1
policy-map set-de
class data
set fr-de
interface Serial0/0/0
encapsulation frame-relay
interface Serial0/0/0.1 point-to-point
ip address 192.168.249.194 255.255.255.252
```

```
frame-relay interface-dlci 100
service output set-de
```

Example Matching Frame Relay DE Bit on the Cisco 7200 and 7500 Series Routers

The following example shows how to configure the service policy called match-de and attach it to an interface. In this example, the class map called data evaluates all packets entering the interface for a DE bit setting of 1. If the entering packet has been a DE bit value of 1, the packet's EXP bit setting is set to 3.

```
class-map data
match fr-de
policy-map match-de
class data
set mpls exp 3
ip routing
ip cef distributed
mpls label protocol ldp
interface Loopback0
 ip address 10.20.20.20 255.255.255.255
interface Ethernet1/0/0
 ip address 10.0.0.2 255.255.255.0
 mpls ip
interface Serial4/0/0
 encapsulation frame-relay
service input match-de
connect 100 Serial4/0/0 100 l2transport
xconnect 10.10.10.10 100 encapsulation mpls
```

Example ATM over MPLS

The table below shows the configuration of ATM over MPLS on two PE routers.

Table 8: ATM over MPLS Configuration Example

PE1	PE2

mpls label protocol ldp
 mpls ldp router-id Loopback0 force
!
interface Loopback0
 ip address 10.16.12.12 255.255.255.255
!
 	
interface ATM4/0
 pvc 0/100 l2transport
  encapsulation aal0
  xconnect 10.13.13.13 100 encapsulation mpls
!
 	
interface ATM4/0.300 point-to-point
 no ip directed-broadcast
 no atm enable-ilmi-trap
 pvc 0/300 l2transport
  encapsulation aal0
  xconnect 10.13.13.13 300 encapsulation mpls
	mpls label protocol ldp
	 mpls ldp router-id Loopback0 force
	!
	interface Loopback0
	 ip address 10.13.13.13 255.255.255.255
	
	interface ATM4/0
	  pvc 0/100 l2transport
	  encapsulation aal0
	  xconnect 10.16.12.12 100 encapsulation mpls
	!
	interface ATM4/0.300 point-to-point
	 no ip directed-broadcast
	 no atm enable-ilmi-trap
	 pvc 0/300 l2transport
	  encapsulation aal0
	  xconnect 10.16.12.12 300 encapsulation mpls

Example Ethernet over MPLS with MPLS Traffic Engineering Fast Reroute

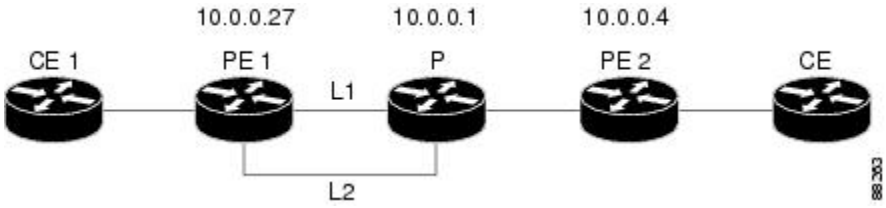
The following configuration example and the figure below show the configuration of Ethernet over MPLS with fast reroute on AToM PE routers.

Routers PE1 and PE2 have the following characteristics:

- A TE tunnel called Tunnel41 is configured between PE1and PE2, using an explicit path through a link called L1. AToM VCs are configured to travel through the FRR-protected tunnel Tunnel41.
- The link L1 is protected by FRR, the backup tunnel is Tunnel1.

- PE2 is configured to forward the AToM traffic back to PE1 through the L2 link.

Figure 2: Fast Reroute Configuration



PE1 Configuration

```
mpls label protocol ldp
mpls traffic-eng tunnels
mpls ldp router-id Loopback1 force
!
pseudowire-class T41
 encapsulation mpls
 preferred-path interface Tunnel41 disable-fallback
!
pseudowire-class IP1
 encapsulation mpls
 preferred-path peer 10.4.0.1 disable-fallback
!
interface Loopback1
 ip address 10.0.0.27 255.255.255.255
!
interface Tunnel1
 ip unnumbered Loopback1
 tunnel destination 10.0.0.1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 10000
 tunnel mpls traffic-eng path-option 1 explicit name FRR
!
interface Tunnel41
 ip unnumbered Loopback1
 tunnel destination 10.0.0.4
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 1000
 tunnel mpls traffic-eng path-option 1 explicit name name-1
 tunnel mpls traffic-eng fast-reroute
!
interface POS0/0
 description pe1name POS8/0/0
 ip address 10.1.0.2 255.255.255.252
 mpls traffic-eng tunnels
 mpls traffic-eng backup-path Tunnel1
 crc 16
 clock source internal
 pos ais-shut
 pos report lrdi
 ip rsvp bandwidth 155000 155000
!
interface POS0/3
 description pe1name POS10/1/0
 ip address 10.1.0.14 255.255.255.252
 mpls traffic-eng tunnels
 crc 16
 clock source internal
 ip rsvp bandwidth 155000 155000
!
interface gigabitethernet3/0.1
 encapsulation dot1Q 203
```

```
 xconnect 10.0.0.4 2 pw-class IP1
!
interface gigabitethernet3/0.2
 encapsulation dot1Q 204
 xconnect 10.0.0.4 4 pw-class T41
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng area 0
!
ip classless
ip route 10.4.0.1 255.255.255.255 Tunnel41
!
ip explicit-path name xxxx-1 enable
 next-address 10.4.1.2
 next-address 10.1.0.10
```

P Configuration

```
ip cef
mpls traffic-eng tunnels
!
interface Loopback1
 ip address 10.0.0.1 255.255.255.255
!
interface FastEthernet1/0/0
 ip address 10.4.1.2 255.255.255.0
 mpls traffic-eng tunnels
 ip rsvp bandwidth 10000 10000
!
interface POS8/0/0
 description xxxx POS0/0
 ip address 10.1.0.1 255.255.255.252
 mpls traffic-eng tunnels
 pos ais-shut
 pos report lrdi
 ip rsvp bandwidth 155000 155000
!
interface POS10/1/0
 description xxxx POS0/3
 ip address 10.1.0.13 255.255.255.252
 mpls traffic-eng tunnels
 ip rsvp bandwidth 155000 155000
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng area 0
```

PE2 Configuration

```
ip cef
mpls label protocol ldp
mpls traffic-eng tunnels
mpls ldp router-id Loopback1 force
!
interface Loopback1
 ip address 10.0.0.4 255.255.255.255
!
interface loopback 2
ip address 10.4.0.1 255.255.255.255
!
interface Tunnel27
 ip unnumbered Loopback1
 tunnel destination 10.0.0.27
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
```

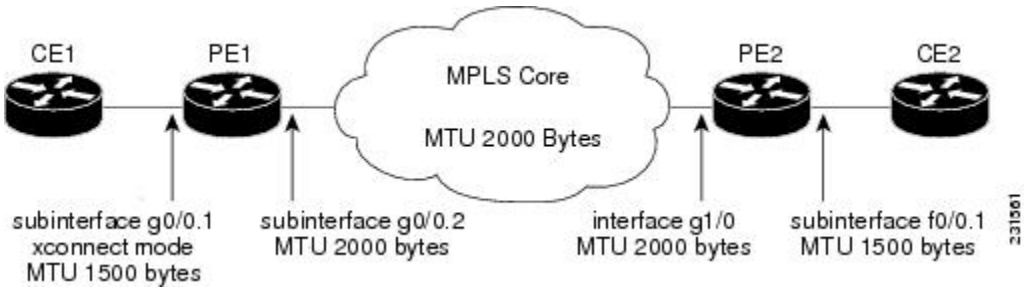
```
 tunnel mpls traffic-eng bandwidth 1000
 tunnel mpls traffic-eng path-option 1 explicit name xxxx-1
!
interface FastEthernet0/0.2
 encapsulation dot1Q 203
 xconnect 10.0.0.27 2 encapsulation mpls
!
interface FastEthernet0/0.3
 encapsulation dot1Q 204
 xconnect 10.0.0.27 4 encapsulation mpls
!
interface FastEthernet1/1
 ip address 10.4.1.1 255.255.255.0
 mpls traffic-eng tunnels
 ip rsvp bandwidth 10000 10000
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng area 0
!
ip explicit-path name xxxx-1 enable
 next-address 10.4.1.2
 next-address 10.1.0.10
```

Example Configuring per-Subinterface MTU for Ethernet over MPLS

The figure below shows a configuration that enables matching MTU values between VC endpoints.

As shown in the figure below, PE1 is configured in xconnect subinterface configuration mode with an MTU value of 1500 bytes in order to establish an end-to-end VC with PE2, which also has an MTU value of 1500 bytes. If PE1 was not set with an MTU value of 1500 bytes, in xconnect subinterface configuration mode, the subinterface would inherit the MTU value of 2000 bytes set on the interface. This would cause a mismatch in MTU values between the VC endpoints, and the VC would not come up.

Figure 3: Configuring MTU Values in xconnect Subinterface Configuration Mode



The following examples show the router configurations in the figure above:

CE1 Configuration

```
interface gigabitethernet0/0
 mtu 1500
 no ip address
!
interface gigabitethernet0/0.1
 encapsulation dot1Q 100
 ip address 10.181.182.1 255.255.255.0
```

PE1 Configuration

```
interface gigabitethernet0/0
 mtu 2000
 no ip address
!
interface gigabitethernet0/0.1
 encapsulation dot1Q 100
 xconnect 10.1.1.152 100 encapsulation mpls
  mtu 1500
!
interface gigabitethernet0/0.2
 encapsulation dot1Q 200
 ip address 10.151.100.1 255.255.255.0
 mpls ip
```

PE2 Configuration

```
interface gigabitethernet1/0
 mtu 2000
 no ip address
!
interface gigabitethernet1/0.2
 encapsulation dot1Q 200
 ip address 10.100.152.2 255.255.255.0
 mpls ip
!
interface fastethernet0/0
 no ip address
!
interface fastethernet0/0.1
 description default MTU of 1500 for FastEthernet
 encapsulation dot1Q 100
 xconnect 10.1.1.151 100 encapsulation mpls
```

CE2 Configuration

```
interface fastethernet0/0
 no ip address
interface fastethernet0/0.1
 encapsulation dot1Q 100
 ip address 10.181.182.2 255.255.255.0
```

The **show mpls l2transport binding**command, issued from router PE1, shows a matching MTU value of 1500 bytes on both the local and remote routers:

```
Router# show mpls l2transport binding
Destination Address: 10.1.1.152,  VC ID: 100
    Local Label: 100
        Cbit: 1,    VC Type: Ethernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
              CV Type: LSPV [2]
    Remote Label: 202
        Cbit: 1,    VC Type: Ethernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: RA [2]
              CV Type: LSPV [2]


Router# show mpls l2transport vc detail
Local interface: Gi0/0.1 up, line protocol up, Eth VLAN 100 up
  Destination address: 10.1.1.152, VC ID: 100, VC status: up
    Output interface: Gi0/0.2, imposed label stack {202}
    Preferred path: not configured
    Default path: active
    Next hop: 10.151.152.2
  Create time: 1d11h, last status change time: 1d11h
```

```
  Signaling protocol: LDP, peer 10.1.1.152:0 up
    Targeted Hello: 10.1.1.151(LDP Id) -> 10.1.1.152
    MPLS VC labels: local 100, remote 202
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 41, send 39
    byte totals:   receive 4460, send 5346
    packet drops:  receive 0, send 0
```

In the following example, you are specifying an MTU of 1501 in xconnect subinterface configuration mode, and that value is out of range, the router enters the command in subinterface configuration mode, where it is accepted:

```
Router# configure terminal
router(config)# interface gigabitethernet0/2.1
router(config-subif)# xconnect 10.10.10.1 100 encapsulation mpls
router(config-subif-xconn)# mtu ?
<64 - 1500> MTU size in bytes
router(config-subif-xconn)# mtu 1501
router(config-subif)# mtu ?
<64 - 17940> MTU size in bytes
```

If the MTU value is not accepted in either xconnect subinterface configuration mode or subinterface configuration mode, then the command is rejected, as shown in the following example:

```
Router# configure terminal
router(config)# interface gigabitethernet0/2.1
router(config-subif)# xconnect 10.10.10.1 100 encapsulation mpls
router(config-subif-xconn)# mtu ?
<64 - 1500> MTU size in bytes
router(config-subif-xconn)# mtu 63
% Invalid input detected at ^ marker
```

Example Configuring MTU Values in xconnect Configuration Mode for L2VPN Interworking

The following example shows an L2VPN Interworking example. The PE1 router has a serial interface configured with an MTU value of 1492 bytes. The PE2 router uses xconnect configuration mode to set a matching MTU of 1492 bytes, which allows the two routers to form an interworking VC. If the PE2 router did not set the MTU value in xconnect configuration mode, the interface would be set to 1500 bytes by default and the VC would not come up.

PE1 Configuration

```
pseudowire-class atom-ipiw
 encapsulation mpls
 interworking ip
!
interface Loopback0
 ip address 10.1.1.151 255.255.255.255
!
interface Serial2/0
 mtu 1492
 no ip address
 encapsulation ppp
 no fair-queue
 serial restart-delay 0
 xconnect 10.1.1.152 123 pw-class atom-ipiw
!
interface Serial4/0
```

```
 ip address 10.151.100.1 255.255.255.252
 encapsulation ppp
 mpls ip
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.151 0.0.0.0 area 0
 network 10.151.100.0 0.0.0.3 area 0
!
mpls ldp router-id Loopback0
```

PE2 Configuration

```
pseudowire-class atom-ipiw
 encapsulation mpls
 interworking ip
!
interface Loopback0
 ip address 10.1.1.152 255.255.255.255
!
interface Ethernet0/0
 no ip address
 xconnect 10.1.1.151 123 pw-class atom-ipiw
  mtu 1492
!
interface Serial4/0
 ip address 10.100.152.2 255.255.255.252
 encapsulation ppp
 mpls ip
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.152 0.0.0.0 area 0
 network 10.100.152.0 0.0.0.3 area 0
!
mpls ldp router-id Loopback0
```

The **show mpls l2transport binding**command shows that the MTU value for the local and remote routers is 1492 bytes.

PE1 Configuration

```
Router# show mpls l2transport binding

Destination Address: 10.1.1.152,  VC ID: 123
    Local Label: 105
        Cbit: 1,    VC Type: PPP,    GroupID: 0
        MTU: 1492,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
              CV Type: LSPV [2]
    Remote Label: 205
        Cbit: 1,    VC Type: Ethernet,    GroupID: 0
        MTU: 1492,   Interface Desc: n/a
        VCCV: CC Type: RA [2]
              CV Type: LSPV [2]
Router# show mpls l2transport vc detail
Local interface: Se2/0 up, line protocol up, PPP up
  MPLS VC type is PPP, interworking type is IP
  Destination address: 10.1.1.152, VC ID: 123, VC status: up
    Output interface: Se4/0, imposed label stack {1003 205}
    Preferred path: not configured
    Default path: active
    Next hop: point2point
  Create time: 00:25:29, last status change time: 00:24:54
  Signaling protocol: LDP, peer 10.1.1.152:0 up
    Targeted Hello: 10.1.1.151(LDP Id) -> 10.1.1.152
    Status TLV support (local/remote)   : enabled/supported
```

```
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: no fault
      Last local SSS circuit status rcvd: no fault
      Last local SSS circuit status sent: no fault
      Last local  LDP TLV    status sent: no fault
      Last remote LDP TLV    status rcvd: no fault
    MPLS VC labels: local 105, remote 205
    Group ID: local n/a, remote 0
    MTU: local 1492, remote 1492
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 30, send 29
    byte totals:   receive 2946, send 3364
    packet drops:  receive 0, send 0
```

PE2 Configuration

```
Router# show mpls l2transport binding

Destination Address: 10.1.1.151,  VC ID: 123
    Local Label: 205
        Cbit: 1,    VC Type: Ethernet,    GroupID: 0
        MTU: 1492,   Interface Desc: n/a
        VCCV: CC Type: RA [2]
              CV Type: LSPV [2]
    Remote Label: 105
        Cbit: 1,    VC Type: Ethernet,    GroupID: 0
        MTU: 1492,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
              CV Type: LSPV [2]
Router# show mpls l2transport vc detail
Local interface: Et0/0 up, line protocol up, Ethernet up
  MPLS VC type is Ethernet, interworking type is IP
  Destination address: 10.1.1.151, VC ID: 123, VC status: up
    Output interface: Se4/0, imposed label stack {1002 105}
    Preferred path: not configured
    Default path: active
    Next hop: point2point
  Create time: 00:25:19, last status change time: 00:25:19
  Signaling protocol: LDP, peer 10.1.1.151:0 up
    Targeted Hello: 10.1.1.152(LDP Id) -> 10.1.1.151
    Status TLV support (local/remote)   : enabled/supported
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: no fault
      Last local SSS circuit status rcvd: no fault
      Last local SSS circuit status sent: no fault
      Last local  LDP TLV    status sent: no fault
      Last remote LDP TLV    status rcvd: no fault
    MPLS VC labels: local 205, remote 105
    Group ID: local n/a, remote 0
    MTU: local 1492, remote 1492
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 29, send 30
    byte totals:   receive 2900, send 3426
    packet drops:  receive 0, send 0
```

Example Removing a Pseudowire

The following example shows how to remove all xconnects:

```
Router# clear xconnect all
02:13:56: Xconnect[ac:Et1/0.1(Eth VLAN)]: provisioning fwder with fwd_type=1, sss_role=1
02:13:56: Xconnect[mpls:10.1.1.2:1234000]: provisioning fwder with fwd_type=2, sss_role=2
02:13:56: Xconnect[ac:Et1/0.2(Eth VLAN)]: provisioning fwder with fwd_type=1, sss_role=2
```

```
02:13:56: Xconnect[mpls:10.1.1.2:1234001]: provisioning fwder with fwd_type=2, sss_role=1
02:13:56: Xconnect[ac:Et1/0.3(Eth VLAN)]: provisioning fwder with fwd_type=1, sss_role=1
02:13:56: Xconnect[mpls:10.1.2.2:1234002]: provisioning fwder with fwd_type=2, sss_role=2
02:13:56: Xconnect[ac:Et1/0.4(Eth VLAN)]: provisioning fwder with fwd_type=1, sss_role=2
02:13:56: Xconnect[mpls:10.1.2.2:1234003]: provisioning fwder with fwd_type=2, sss_role=1
02:13:56: MPLS peer 10.1.1.2 vcid 1234000, VC DOWN, VC state DOWN
02:13:56: MPLS peer 10.1.1.2 vcid 1234001, VC DOWN, VC state DOWN
02:13:56: MPLS peer 10.1.2.2 vcid 1234002, VC DOWN, VC state DOWN
02:13:56: MPLS peer 10.1.2.2 vcid 1234003, VC DOWN, VC state DOWN
02:13:56: XC AUTH [Et1/0.1, 1001]: Event: start xconnect authorization, state changed from
 IDLE to AUTHORIZING
02:13:56: XC AUTH [Et1/0.1, 1001]: Event: found xconnect authorization, state changed from
 AUTHORIZING to DONE
02:13:56: XC AUTH [Et1/0.3, 1003]: Event: start xconnect authorization, state changed from
 IDLE to AUTHORIZING
02:13:56: XC AUTH [Et1/0.3, 1003]: Event: found xconnect authorization, state changed from
 AUTHORIZING to DONE
02:13:56: XC AUTH [10.1.1.2, 1234001]: Event: start xconnect authorization, state changed
from IDLE to AUTHORIZING
02:13:56: XC AUTH [10.1.1.2, 1234001]: Event: found xconnect authorization, state changed
from AUTHORIZING to DONE
02:13:56: XC AUTH [10.1.2.2, 1234003]: Event: start xconnect authorization, state changed
from IDLE to AUTHORIZING
02:13:56: XC AUTH [10.1.2.2, 1234003]: Event: found xconnect authorization, state changed
from AUTHORIZING to DONE
02:13:56: XC AUTH [Et1/0.1, 1001]: Event: free xconnect authorization request, state changed
 from DONE to END
02:13:56: XC AUTH [Et1/0.3, 1003]: Event: free xconnect authorization request, state changed
 from DONE to END
02:13:56: XC AUTH [10.1.1.2, 1234001]: Event: free xconnect authorization request, state
changed from DONE to END
02:13:56: XC AUTH [10.1.2.2, 1234003]: Event: free xconnect authorization request, state
changed from DONE to END
02:13:56: MPLS peer 10.1.1.2 vcid 1234001, VC UP, VC state UP
02:13:56: MPLS peer 10.1.2.2 vcid 1234003, VC UP, VC state UP
02:13:56: MPLS peer 10.1.1.2 vcid 1234000, VC UP, VC state UP
02:13:56: MPLS peer 10.1.2.2 vcid 1234002, VC UP, VC state UP
```

The following example shows how to remove all the xconnects associated with peer router 10.1.1.2:

```
Router# clear xconnect peer 10.1.1.2 all
02:14:08: Xconnect[ac:Et1/0.1(Eth VLAN)]: provisioning fwder with fwd_type=1, sss_role=1
02:14:08: Xconnect[mpls:10.1.1.2:1234000]: provisioning fwder with fwd_type=2, sss_role=2
02:14:08: Xconnect[ac:Et1/0.2(Eth VLAN)]: provisioning fwder with fwd_type=1, sss_role=2
02:14:08: Xconnect[mpls:10.1.1.2:1234001]: provisioning fwder with fwd_type=2, sss_role=1
02:14:08: MPLS peer 10.1.1.2 vcid 1234000, VC DOWN, VC state DOWN
02:14:08: MPLS peer 10.1.1.2 vcid 1234001, VC DOWN, VC state DOWN
02:14:08: XC AUTH [Et1/0.1, 1001]: Event: start xconnect authorization, state changed from
 IDLE to AUTHORIZING
02:14:08: XC AUTH [Et1/0.1, 1001]: Event: found xconnect authorization, state changed from
 AUTHORIZING to DONE
02:14:08: XC AUTH [10.1.1.2, 1234001]: Event: start xconnect authorization, state changed
from IDLE to AUTHORIZING
02:14:08: XC AUTH [10.1.1.2, 1234001]: Event: found xconnect authorization, state changed
from AUTHORIZING to DONE
02:14:08: XC AUTH [Et1/0.1, 1001]: Event: free xconnect authorization request, state changed
 from DONE to END
02:14:08: XC AUTH [10.1.1.2, 1234001]: Event: free xconnect authorization request, state
changed from DONE to END
02:14:08: MPLS peer 10.1.1.2 vcid 1234001, VC UP, VC state UP
02:14:08: MPLS peer 10.1.1.2 vcid 1234000, VC UP, VC state UP
```

The following example shows how to remove the xconnects associated with peer router 10.1.1.2 and VC ID 1234001:

```
Router# clear xconnect peer 10.1.1.2 vcid 1234001
02:14:23: Xconnect[ac:Et1/0.2(Eth VLAN)]: provisioning fwder with fwd_type=1, sss_role=1
02:14:23: Xconnect[mpls:10.1.1.2:1234001]: provisioning fwder with fwd_type=2, sss_role=2
02:14:23: MPLS peer 10.1.1.2 vcid 1234001, VC DOWN, VC state DOWN
02:14:23: XC AUTH [Et1/0.2, 1002]: Event: start xconnect authorization, state changed from
 IDLE to AUTHORIZING
02:14:23: XC AUTH [Et1/0.2, 1002]: Event: found xconnect authorization, state changed from
```

```
 AUTHORIZING to DONE
02:14:23: XC AUTH [Et1/0.2, 1002]: Event: free xconnect authorization request, state changed
 from DONE to END
02:14:23: MPLS peer 10.1.1.2 vcid 1234001, VC UP, VC state UP
```

The following example shows how to remove the xconnects associated with interface Ethernet 1/0.1:

```
Router# clear xconnect interface eth1/0.1

02:14:48: Xconnect[ac:Et1/0.1(Eth VLAN)]: provisioning fwder with fwd_type=1, sss_role=2
02:14:48: Xconnect[mpls:10.1.1.2:1234000]: provisioning fwder with fwd_type=2, sss_role=1
02:14:48: MPLS peer 10.1.1.2 vcid 1234000, VC DOWN, VC state DOWN
02:14:48: XC AUTH [10.1.1.2, 1234000]: Event: start xconnect authorization, state changed
from IDLE to AUTHORIZING
02:14:48: XC AUTH [10.1.1.2, 1234000]: Event: found xconnect authorization, state changed
from AUTHORIZING to DONE
02:14:48: XC AUTH [10.1.1.2, 1234000]: Event: free xconnect authorization request, state
changed from DONE to END
```

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*
Any Transport over MPLS	"Overview" section of Cisco Any Transport over MPLS
Any Transport over MPLS for the Cisco 10000 series router	Cisco 10000 Series Router Broadband Aggregation, Leased-Line, and MPLS Configuration Guide
Layer 2 Tunnel Protocol Version 3 (L2TPv3)	Layer 2 Tunnel Protocol Version 3 (L2TPv3)
L2VPN interworking	L2VPN Interworking

Standards

Standard	Title
draft-martini-l2circuit-trans-mpls-08.txt	*Transport of Layer 2 Frames Over MPLS*
draft-martini-l2circuit-encap-mpls-04.txt	*Encapsulation Methods for Transport of Layer 2 Frames Over MPLS*

MIBs

MIB	MIBs Link
ATM AAL5 over MPLS and ATM Cell Relay over MPLS: • MPLS LDP MIB (MPLS-LDP-MIB.my) • ATM MIB (ATM-MIB.my) • CISCO AAL5 MIB (CISCO-AAL5-MIB.my) • Cisco Enterprise ATM Extension MIB (CISCO-ATM-EXT-MIB.my) • Supplemental ATM Management Objects (CISCO-IETF-ATM2-PVCTRAP-MIB.my) • Interfaces MIB (IF-MIB.my) Ethernet over MPLS: • CISCO-ETHERLIKE-CAPABILITIES.my • Ethernet MIB (ETHERLIKE-MIB.my) • Interfaces MIB (IF-MIB.my) • MPLS LDP MIB (MPLS-LDP-MIB.my) Frame Relay over MPLS: • Cisco Frame Relay MIB (CISCO-FRAME-RELAY-MIB.my) • Interfaces MIB (IF-MIB.my) • MPLS LDP MIB (MPLS-LDP-MIB.my) HDLC and PPP over MPLS: • MPLS LDP MIB (MPLS-LDP-MIB.my) • Interfaces MIB (IF-MIB.my)	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 3032	*MPLS Label Stack Encoding*
RFC 3036	*LDP Specification*

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 Any Transport 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 Any Transport over MPLS

Feature Name	Releases	Feature Information
Any Transport over MPLS	12.0(10)ST 12.0(21)ST 12.0(22)S 12.0(23)S 12.0(25)S 12.0(26)S 12.0(27)S 12.0(29)S 12.0(30)S 12.0(31)S 12.0(32)S 12.1(8a)E 12.2(14)S 12.2(15)T 12.2(28)SB 12.2(33)SRB 12.2(33)SXH 12.2(33)SRC 12.2(33)SRD 12.2(1)SRE 12.4(11)T 15.0(1)S 15.1(3)S	

Feature Name	Releases	Feature Information
		In Cisco IOS Release 12.0(10)ST, Any Transport over MPLS: ATM AAL5 over MPLS was introduced on the Cisco 12000 series routers. In Cisco IOS Release 12.1(8a)E, Ethernet over MPLS was introduced on the Cisco 7600 series Internet router. In Cisco IOS Release 12.0(21)ST, Any Transport over MPLS: Ethernet over MPLS was introduced on the Cisco 12000 series routers. ATM AAL5 over MPLS was updated. In Cisco IOS Release 12.0(22)S, Ethernet over MPLS was integrated into this release. Support for the Cisco 10720 Internet router was added. ATM AAL5 over MPLS was integrated into this release for the Cisco 12000 series routers. In Cisco IOS Release 12.0(23)S, the following new features were introduced and support was added for them on the Cisco 7200 and 7500 series routers: • ATM Cell Relay over MPLS (single cell relay, VC mode) • Frame Relay over MPLS • HDLC over MPLS • PPP over MPLS Cisco IOS Release 12.0(23)S also added support on the Cisco 12000, 7200, and 7500 series routers for the following features: • ATM AAL5 over MPLS • Ethernet over MPLS (VLAN mode) The AToM features were integrated into Cisco IOS Release 12.2(14)S. The AToM features were

Feature Name	Releases	Feature Information
		integrated into Cisco IOS Release 12.2(15)T. In Cisco IOS Release 12.0(25)S, the following new features were introduced: • New commands for configuring AToM • Ethernet over MPLS: port mode • ATM Cell Relay over MPLS: packed cell relay • ATM Cell Relay over MPLS: VP mode • ATM Cell Relay over MPLS: port mode • Distributed Cisco Express Forwarding mode for Frame Relay, PPP, and HDLC over MPLS • Fast reroute with AToM • Tunnel selection • Traffic policing • QoS support

Feature Name	Releases	Feature Information

Feature Name	Releases	Feature Information
		In Cisco IOS Release 12.0(26)S, the following new features were introduced: • Support for connecting disparate attachment circuits. See L2VPN Interworking for more information. • QoS functionality with AToM for the Cisco 7200 series routers. Support for FECN and BECN marking with Frame Relay over MPLS. (See BECN and FECN Marking for Frame Relay over MPLS for more information.) In Cisco IOS Release 12.0(27)S, the following new features were introduced: • ATM Cell Relay over MPLS: Packed Cell Relay for VC, PVP, and port mode for the Cisco 12000 series router. • Support for ATM over MPLS on the Cisco 12000 series 4-port OC-12X/STM-4 ATM ISE line card. This feature was integrated into Cisco IOS Release 12.2(25)S for the Cisco 7200 and 7500 series routers. In Cisco IOS Release 12.0(29)S, the “Any Transport over MPLS Sequencing Support” feature was added for the Cisco 7200 and 7500 series routers. In Cisco IOS Release 12.0(30)S, the following new features were introduced: In Cisco IOS Release 12.0(31)S, the Cisco 12000 series router introduced the following enhancements:

Feature Name	Releases	Feature Information
		• AToM VC Independence--With this enhancement, fast reroute is accomplished in less than 50 milliseconds, regardless of the number of VCs configured. • Support for ISE line cards on the 2.5G ISE SPA Interface Processor (SIP). In Cisco IOS Release 12.0(32)S, the Cisco 12000 series router added engine 5 line card support for the following transport types: • Ethernet over MPLS • Frame Relay over MPLS • HDLC over MPLS • PPP over MPLS

Feature Name	Releases	Feature Information

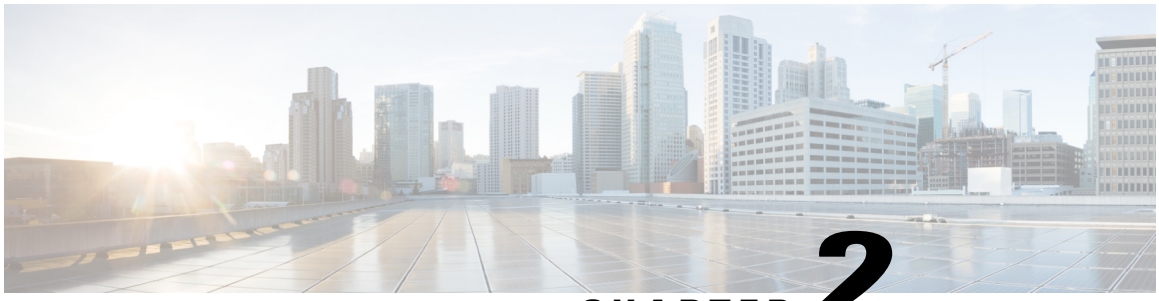
Feature Name	Releases	Feature Information
		This feature was integrated into Cisco IOS Release 12.2(28)SB on the Cisco 10000 series routers. Platform-specific configuration information is contained in the "Configuring Any Transport over MPLS" section of the Cisco 10000 Series Router Broadband Aggregation, Leased-Line, and MPLS Configuration Guide. Any Transport over MPLS was integrated into Cisco IOS Release 12.4(11)T with support for the following features: • Any Transport over MPLS: Ethernet over MPLS: Port Mode • Any Transport over MPLS: Ethernet over MPLS: VLAN Mode • Any Transport over MPLS: Ethernet over MPLS: VLAN ID Rewrite • Any Transport over MPLS: Frame Relay over MPLS • Any Transport over MPLS: AAL5 over MPLS • Any Transport over MPLS: ATM OAM Emulation This feature was integrated into Cisco IOS Release 12.2(33)SRB to support the following features on the Cisco 7600 router: • Any Transport over MPLS: Frame Relay over MPLS • Any Transport over MPLS: ATM Cell Relay over MPLS: Packed Cell Relay • Any Transport over MPLS: Ethernet over MPLS • AToM Static Pseudowire Provisioning

Feature Name	Releases	Feature Information
		Platform-specific configuration information is contained in the following documents: • The "Configuring PFC3BXL and PFC3B Mode Multiprotocol Label Switching" module of the Cisco 7600 Series Cisco IOS Software Configuration Guide, Release 12.2SR • The "Configuring Multiprotocol Label Switching on the Optical Services Modules" module of the OSM Configuration Note, Release 12.2SR • The "Configuring Multiprotocol Label Switching on FlexWAN and Enhanced FlexWAN Modules" module of the FlexWAN and Enhanced FlexWAN Modules Configuration Guide • The "Configuring Any Transport over MPLS on a SIP" section of the Cisco 7600 Series Router SIP, SSC, and SPA Software Configuration Guide • The "Configuring AToM VP Cell Mode Relay Support" section of the Cisco 7600 Series Router SIP, SSC, and SPA Software Configuration Guide • The *Cross-Platform Release Notes for Cisco IOS Release 12.2SR for the Cisco 7600 Series Routers*

Feature Name	Releases	Feature Information

Feature Name	Releases	Feature Information
		This feature was integrated into Cisco IOS Release 12.2(33)SXH and supports the following features: • Any Transport over MPLS: Ethernet over MPLS: Port Mode • Any Transport over MPLS: AAL5 over MPLS • Any Transport over MPLS: ATM OAM Emulation • Any Transport over MPLS: Single Cell Relay--VC Mode • Any Transport over MPLS: ATM Cell Relay over MPLS--VP Mode • Any Transport over MPLS: Packed Cell Relay--VC/VP Mode • Any Transport over MPLS: Ethernet over MPLS • ATM Port Mode Packed Cell Relay over AToM • AToM Tunnel Selection The following features were integrated into Cisco IOS Release 12.2(33)SRC: • AToM Tunnel Selection for the Cisco 7200 and Cisco 7300 routers • Per-Subinterface MTU for Ethernet over MPLS (EoMPLS) In Cisco IOS Release 12.2(33)SRD, support for ATM Cell Relay over MPLS in port mode on Cisco 7600 series routers was added. Per Subinterface MTU for Ethernet over MPLS (EoMPLS) was integrated into Cisco IOS Release

Feature Name	Releases	Feature Information
		15.1(3)S.
MPLS L2VPN Clear Xconnect Command	12.2(1)SRE 15.0(1)S	These features are supported on Cisco 7600 routers in Cisco IOS Release 12.2(1)SRE and Cisco IOS Release 15.0(1)S. These features enable you to: • Reset a VC associated with an interface, a peer address, or on all the configured xconnect circuit attachments • Set the control word on dynamic pseudowires. • Enable ATM cell packing for static pseudowires. The following commands were introduced or modified by these features: **cell-packing**, **clear xconnect**, **control-word**, **encapsulation (Any Transport over MPLS)**, **oam-ac emulation-enable**.
MPLS MTU Command for GRE Tunnels	15.1(1)T 15.1(2)S	This feature allows you to reset the MPLS MTU size in GRE tunnels from default to the maximum. The **maximum** keyword was replaced with the **max** keyword. The following command was modified by this feature: **mpls mtu**.
ATM Port mode Packed Cell Relay over MPLS	15.2(1)S	This feature was integrated into Cisco IOS Release 12.2(1)S.
Any Transport over MPLS (AToM): ATM Cell Relay over MPLS: Packed Cell Relay	15.2(1)S	This feature was integrated into Cisco IOS Release 12.2(1)S.



CHAPTER 2

L2VPN Interworking

Layer 2 Virtual Private Network (L2VPN) Interworking allows you to connect disparate attachment circuits. This feature module explains how to configure the following L2VPN Interworking features:

- Ethernet/VLAN to ATM AAL5 Interworking
- Ethernet/VLAN to Frame Relay Interworking
- Ethernet/VLAN to PPP Interworking
- Ethernet to VLAN Interworking
- Frame Relay to ATM AAL5 Interworking
- Frame Relay to PPP Interworking
- Ethernet/VLAN to ATM virtual channel identifier (VPI) and virtual channel identifier (VCI) Interworking
- L2VPN Interworking: VLAN Enable/Disable Option for AToM

Finding Feature Information

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

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

Prerequisites for L2VPN Interworking

Before you configure L2VPN Interworking on a router:

- You must enable Cisco Express Forwarding.
- On the Cisco 12000 series Internet router, before you configure Layer 2 Tunnel Protocol version 3 (L2TPv3) for L2VPN Interworking on an IP Services Engine (ISE/Engine 3) or Engine 5 interface, you must also enable the L2VPN feature bundle on the line card.

To enable the feature bundle, enter the **hw-module slot np mode feature** command in global configuration mode as follows:

```
Router# configure terminal
Router(config)# hw-module slot slot-number np mode feature
```

Restrictions for L2VPN Interworking

General Restrictions

This section lists general restrictions that apply to L2VPN Interworking. Other restrictions that are platform-specific or device-specific are listed in the following sections.

- The interworking type on one provider edge (PE) router must match the interworking type on the peer PE router.
- The following quality of service (QoS) features are supported with L2VPN Interworking:
  - Static IP type of service (ToS) or Multiprotocol Label Switching (MPLS) experimental bit (EXP) setting in tunnel header
  - IP ToS reflection in tunnel header (Layer 2 Tunnel Protocol Version 3 (L2TPv3) only)
  - Frame Relay policing
  - Frame Relay data-link connection identifier (DLCI)-based congestion management (Cisco 7500/Versatile Interface Processor (VIP))
  - One-to-one mapping of VLAN priority bits to MPLS EXP bits
- Only ATM AAL5 VC mode is supported; ATM VP and port mode are not supported.
- In Cisco IOS Release 12.2(52)SE and Cisco IOS Release 12.2(33)SRE, the **encapsulation** command supports only the **mpls** keyword. The **l2tpv3** keyword is not supported. The **interworking** command supports only the **ethernet** and **vlan** keywords. The **ip** keyword is not supported.

Cisco 7600 Series Routers Restrictions

The following line cards are supported on the Cisco 7600 series router. The first table below shows the line cards that are supported on the WAN (ATM, Frame Relay, or PPP) side of the interworking link. The second table below shows the line cards that are supported on the Ethernet side of the interworking link. For more details on the Cisco 7600 routers supported shared port adapters and line cards, see the following document:

- Release Notes for Cisco IOS Release 12.2SR for the Cisco 7600 Series Routers

Table 10: Cisco 7600 Series Routers: Supported Line Cards for the WAN Side

Interworking Type	Core-Facing Line Cards	Customer-Edge Line Cards
Ethernet (bridged) (ATM and Frame Relay)	Any	EflexWAN SIP-200 SIP-400
IP (routed) (ATM, Frame Relay, and PPP)	Any	EflexWAN SIP-200

Table 11: Cisco 7600 Series Routers: Supported Line Cards for the Ethernet Side

Interworking Type	Ethernet over MPLS Mode	Core-Facing Line Cards	Customer-Edge Line Cards
Ethernet (bridged)	Policy feature card (PFC) based	Any, except optical service module (OSM) and ES40	Catalyst LAN SIP-600
Ethernet (bridged)	Switched virtual interface (SVI) based	EflexWAN ES20 ES+40 SIP-200 SIP-400 SIP-600	Catalyst LAN EflexWAN (with MPB) ES20 ES+40 SIP-200 (with MPB) SIP-400 (with MPB) SIP-600
Ethernet (bridged)	Scalable (with E-MPB)	Any, except OSM	ES20 SIP-600 and SIP-400 with Gigabit Ethernet (GE) SPA
IP (routed)	PFC-based	Catalyst LAN SIP-600 **Note:** PFC-based mode is not supported with routed interworking in Cisco IOS Release 12.2(33)SRD. Use SVI, Scalable, or Ethernet virtual connection (EVC) based Ethernet over MPLS (EoMPLS) instead.	Catalyst LAN SIP-600 **Note:** PFC-based mode is not supported with routed interworking in Cisco IOS Release 12.2(33)SRD. Use SVI, Scalable, or EVC-based EoMPLS instead.

Interworking Type	Ethernet over MPLS Mode	Core-Facing Line Cards	Customer-Edge Line Cards
IP (routed)	SVI-based	Any, except Catalyst LAN and OSM.	Catalyst LAN EflexWAN (with MPB) ES20 SIP-200 (with MPB) SIP-400 (with MPB) SIP-600

The following restrictions apply to the Cisco 7600 series routers and L2VPN Interworking:

- OAM Emulation is not required with L2VPN Interworking on the SIP-200, SIP-400, and Flexwan2 line cards.
- Cisco 7600 series routers support the L2VPN Interworking: VLAN Enable/Disable Option for AToM feature starting in Cisco IOS Release 12.2(33)SRE. This feature has the following restrictions:
  - PFC-based EoMPLS is not supported.
  - Scalable and SVI-based EoMPLS are supported with the SIP-400 line card.
- The Cisco 7600 series routers do not support L2VPN Interworking over L2TPv3.
- Cisco 7600 series routers support only the following interworking types:
  - Ethernet/VLAN to Frame Relay (IP and Ethernet modes)
  - Ethernet/VLAN to ATM AAL5SNAP (IP and Ethernet modes)
  - Ethernet/VLAN to PPP (IP only)
  - Ethernet to VLAN Interworking
- Cisco 7600 series routers do not support the following interworking types:
  - Ethernet/VLAN to ATM AAL5MUX
  - Frame Relay to PPP Interworking
  - Frame Relay to ATM AAL5 Interworking
- Both ends of the interworking link must be configured with the same encapsulation and interworking type:
  - If you use Ethernet encapsulation, you must use the Ethernet (bridged) interworking type. If you are not using Ethernet encapsulation, you can use a bridging mechanism, such as routed bridge encapsulation (RBE).
  - If you use an IP encapsulation (such as ATM or Frame Relay), you must use the IP (routed) interworking type. The PE routers negotiate the process for learning and resolving addresses.
  - You must use the same MTU size on the attachment circuits at each end of the pseudowire.
- PFC-based EoMPLS is not supported on ES40 line cards. SVI and EVC/scalable EoMPLS are the alternative options.

- PFC-based EoMPLS is not supported for Routed/IP interworking in Cisco IOS Release 12.2(33)SRD and later releases. The alternative Routed/IP interworking options are SVI and EVC or scalable EoMPLS. However, PFC-based EoMPLS is supported for Ethernet/Bridged interworking and for like-to-like over AToM.

Cisco 12000 Series Router Restrictions

For more information about hardware requirements on the Cisco12000 series routers, see the Cross-Platform Release Notes for Cisco IOS Release 12.0S.

For QOS support on the Cisco 12000 series routers, see Any Transport over MPLS (AToM): Layer 2 QoS (Quality of Service) for the Cisco 12000 Series Router

Frame Relay to PPP and High-Level Data Link Control Interworking

The Cisco 12000 series Internet router does not support L2VPN Interworking with PPP and high-level data link control (HDLC) transport types in Cisco IOS releases earlier than Cisco IOS Release 12.0(32)S.

In Cisco IOS Release 12.0(32)S and later releases, the Cisco 12000 series Internet router supports L2VPN interworking for Frame Relay over MPLS and PPP and HDLC over MPLS only on the following shared port adapters (SPAs):

- ISE/Engine 3 SPAs:
  - SPA-2XCT3/DS0 (2-port channelized T3 to DS0)
  - SPA-4XCT3/DS0 (4-port channelized T3 to DS0)
- Engine 5 SPAs:
  - SPA-1XCHSTM1/OC-3 (1-port channelized STM-1c/OC-3c to DS0)
  - SPA-8XCHT1/E1 (8-port channelized T1/E1)
  - SPA-2XOC-48-POS/RPR (2-port OC-48/STM16 POS/RPR)
  - SPA-OC-192POS-LR (1-port OC-192/STM64 POS/RPR)
  - SPA-OC-192POS-XFP (1-port OC-192/STM64 POS/RPR)

L2VPN Interworking over L2TPv3

On the Cisco 12000 series Internet router, Ethernet (bridged) interworking is not supported for L2TPv3. Only IP (routed) interworking is supported.

IP (routed) interworking is not supported in an L2TPv3 pseudowire that is configured for data sequencing (using the **sequencing** command).

In Cisco IOS Release 12.0(32)SY and later releases, the Cisco 12000 series Internet router supports L2VPN Interworking over L2TPv3 tunnels in IP mode on ISE and Engine 5 line cards as follows:

- On an ISE interface configured for L2TPv3 tunneling, the following Layer 2 encapsulations are supported:
  - ATM adaptation layer type-5 (AAL5)
  - Ethernet

- 802.1q (VLAN)
- Frame Relay DLCI

- On an Engine 5 interface configured for L2TPv3 tunneling, the following Layer 2 encapsulations are supported:
  - Ethernet
  - 802.1q (VLAN)
  - Frame Relay DLCI

For more information, refer to Layer 2 Tunnel Protocol Version 3.

The only frame format supported for L2TPv3 interworking on Engine 5 Ethernet SPAs is Ethernet Version 2 (also known as Ethernet II) with the Ether type 0x0800 value set as Internet Protocol Payload and (optionally) 802.1q VLAN. Ethernet packets with other Ethernet frame formats are dropped.

Remote Ethernet Port Shutdown Support

The Cisco Remote Ethernet Port Shutdown feature (which minimizes potential data loss after a remote link failure) is supported only on the following Engine 5 Ethernet SPAs:

- SPA-8XFE (8-port Fast Ethernet)
- SPA-2X1GE (2-port Gigabit Ethernet)
- SPA-5X1GE (5-port Gigabit Ethernet)
- SPA-10X1GE (10-port Gigabit Ethernet)
- SPA-1X10GE (1-port 10-Gigabit Ethernet)

For more information about this feature, refer to Any Transport over MPLS (AToM): Remote Ethernet Port Shutdown.

L2VPN Any-to-Any Interworking on Engine 5 Line Cards

The table below shows the different combinations of transport types supported for L2VPN interworking on Engine 3 and Engine 5 SPA interfaces connected through an attachment circuit over MPLS or L2TPv3.

Table 12: Engine 3 and Engine 5 Line Cards/SPAs Supported for L2VPN Interworking

Attachment Circuit 1 (AC1)	Attachment Circuit 2 (AC2)	Interworking Mode	AC1 Engine Type and Line Card/SPA	AC2 Engine Type and Line Card/SPA
Frame Relay	Frame Relay	IP	Engine 5 POS and channelized	Engine 3 ATM line cards
Frame Relay	ATM	Ethernet	Engine 5 POS and channelized	Engine 3 ATM line cards
Frame Relay	ATM	IP	Engine 5 POS and channelized	Engine 3 ATM line cards

Attachment Circuit 1 (AC1)	Attachment Circuit 2 (AC2)	Interworking Mode	AC1 Engine Type and Line Card/SPA	AC2 Engine Type and Line Card/SPA
Frame Relay	Ethernet	Ethernet	Engine 5 POS and channelized	Engine 5 Gigabit Ethernet
Frame Relay	Ethernet	IP	Engine 5 POS and channelized	Engine 5 Gigabit Ethernet
Frame Relay	VLAN	Ethernet	Engine 5 POS and channelized	Engine 5 Gigabit Ethernet
Frame Relay	VLAN	IP	Engine 5 POS and channelized	Engine 5 Gigabit Ethernet
Ethernet	Ethernet	Ethernet	Engine 5 Gigabit Ethernet	Engine 5 Gigabit Ethernet
Ethernet	Ethernet	IP	Engine 5 Gigabit Ethernet	Engine 5 Gigabit Ethernet
Ethernet	VLAN	Ethernet	Engine 5 Gigabit Ethernet	Engine 5 Gigabit Ethernet
Ethernet	VLAN	IP	Engine 5 Gigabit Ethernet	Engine 5 Gigabit Ethernet
ATM	Ethernet	Ethernet	Engine 3 ATM line cards	Engine 5 Gigabit Ethernet
ATM	Ethernet	IP	Engine 3 ATM line cards	Engine 5 Gigabit Ethernet

On the Cisco 12000 series Engine 3 line card, Network Layer Protocol ID (NLPID) encapsulation is not supported in routed mode; and neither NLPID nor AAL5MUX is supported in bridged mode.

- On the Cisco 12000 series Internet router, Ethernet (bridged) interworking is not supported for L2TPv3.

In an L2VPN Interworking configuration, after you configure L2TPv3 tunnel encapsulation for a pseudowire using the **encapsulation l2tpv3**command, you cannot enter the **interworking ethernet** command.

- On Ethernet SPAs on the Cisco 12000 series Internet router, the only frame format supported for L2TPv3 interworking is Ethernet Version 2 (also known as Ethernet II) with the Ether type 0x0800 value set as Internet Protocol Payload and [optionally] 802.1q VLAN.

Ethernet packets with other Ethernet frame formats are dropped.

ATM AAL5 Interworking Restrictions

The following restrictions apply to ATM AAL5 Interworking:

- Switched virtual circuits (SVCs) are not supported.
- Inverse Address Resolution Protocol (ARP) is not supported with IP interworking.
- Customer edge (CE) routers must use point-to-point subinterfaces or static maps.
- Both AAL5MUX and AAL5SNAP encapsulation are supported. In the case of AAL5MUX, no translation is needed.
- In the Ethernet end-to-end over ATM scenario, the following translations are supported:
  - Ethernet without LAN frame check sequence (FCS) (AAAA030080C200070000)
  - Spanning tree (AAAA030080c2000E)

Everything else is dropped.

- In the IP over ATM scenario, the IPv4 (AAAA030000000800) translation is supported. Everything else is dropped.
- Operation, Administration, and Management (OAM) emulation for L2VPN Interworking is the same as like-to-like. The end-to-end F5 loopback cells are looped back on the PE router. When the pseudowire is down, an F5 end-to-end segment Alarm Indication Signal (AIS)/Remote Defect Identification (RDI) is sent from the PE router to the CE router.
- Interim Local Management Interface (ILMI) can manage virtual circuits (VCs) and permanent virtual circuits (PVCs).
- To enable ILMI management, configure ILMI PVC 0/16 on the PE router's ATM interface. If a PVC is provisioned or deleted, an ilmiVCCChange trap is sent to the CE router.
- Only the user side of the User-Network Interface (UNI) is supported; the network side of the UNI is not supported.

Ethernet VLAN Interworking Restrictions

The following restrictions apply to Ethernet/VLAN interworking:

- When you configure VLAN to Ethernet interworking, VLAN to Frame Relay (routed), or ATM using Ethernet (bridged) interworking, the PE router on the Ethernet side that receives a VLAN tagged frame from the CE router removes the VLAN tag. In the reverse direction, the PE router adds the VLAN tag to the frame before sending the frame to the CE router.

(If you enable the L2VPN Interworking: VLAN Enable/Disable Option for AToM feature with the **interworking vlan** command, VLAN ID is included as part of the Ethernet frame. See the VLAN Interworking, on page 122 for more information.)

- In bridged interworking from VLAN to Frame Relay, the Frame Relay PE router does not strip off VLAN tags from the Ethernet traffic it receives.
- The Cisco 10720 Internet router supports Ethernet to VLAN Interworking Ethernet only over L2TPv3.

- Ethernet interworking for a raw Ethernet port or a VLAN trunk is not supported. Traffic streams are not kept separate when traffic is sent between transport types.
- In routed mode, only one CE router can be attached to an Ethernet PE router.
- There must be a one-to-one relationship between an attachment circuit and the pseudowire. Point-to-multipoint or multipoint-to-point configurations are not supported.
- Configure routing protocols for point-to-point operation on the CE routers when configuring an Ethernet to non-Ethernet setup.
- In the IP interworking mode, the IPv4 (0800) translation is supported. The PE router captures ARP (0806) packets and responds with its own MAC address (proxy ARP). Everything else is dropped.
- The Ethernet or VLAN must contain only two IP devices: PE router and CE router. The PE router performs proxy ARP and responds to all ARP requests it receives. Therefore, only one CE and one PE router should be on the Ethernet or VLAN segment.
- If the CE routers are doing static routing, you can perform the following tasks:
  - The PE router needs to learn the MAC address of the CE router to correctly forward traffic to it. The Ethernet PE router sends an Internet Control Message Protocol (ICMP) Router discovery protocol (RDP) solicitation message with the source IP address as zero. The Ethernet CE router responds to this solicitation message. To configure the Cisco CE router's Ethernet or VLAN interface to respond to the ICMP RDP solicitation message, issue the **ip irdp**command in interface configuration mode. If you do not configure the CE router, traffic is dropped until the CE router sends traffic toward the PE router.
  - To disable the CE routers from running the router discovery protocol, issue the **ip irdp maxadvertinterval 0** command in interface mode.
- This restriction applies if you configure interworking between Ethernet and VLAN with Catalyst switches as the CE routers. The spanning tree protocol is supported for Ethernet interworking. Ethernet interworking between an Ethernet port and a VLAN supports spanning tree protocol only on VLAN 1. Configure VLAN 1 as a nonnative VLAN.
- When you change the interworking configuration on an Ethernet PE router, clear the ARP entry on the adjacent CE router so that it can learn the new MAC address. Otherwise, you might experience traffic drops.

Restrictions

The following restrictions apply to the L2VPN Interworking: VLAN Enable/Disable Option for AToM feature, which allows the VLAN ID to be included as part of the Ethernet frame:

- The L2VPN Interworking: VLAN Enable/Disable Option for AToM feature is supported on the following releases:
  - Cisco IOS release 12.2(52)SE for the Cisco Catalyst 3750 Metro switches
  - Cisco IOS Release 12.2(33)SRE for the Cisco 7600 series routers
- L2VPN Interworking: VLAN Enable/Disable Option for AToM is not supported with L2TPv3. You can configure the featue only with AToM.

- If the interface on the PE router is a VLAN interface, it is not necessary to specify the **interworking vlan** command on that PE router.
- The L2VPN Interworking: VLAN Enable/Disable Option for AToM feature works only with the following attachment circuit combinations:
  - Ethernet to Ethernet
  - Ethernet to VLAN
  - VLAN to VLAN
- If you specify an interworking type on a PE router, that interworking type must be enforced. The interworking type must match on both PE routers. Otherwise, the VC may be in an incompatible state and remain in the down state. If the attachment circuit (AC) is VLAN, the PE router can negotiate (autosense) the VC type using Label Distribution Protocol (LDP).

For example, both PE1 and PE2 use Ethernet interfaces, and VLAN interworking is specified on PE1 only. PE2 is not configured with an interworking type and cannot autosense the interworking type. The result is an incompatible state where the VC remains in the down state.

On the other hand, if PE1 uses an Ethernet interface and VLAN interworking is enabled (which will enforce VLAN as the VC type), and PE2 uses a VLAN interface and interworking is not enabled (which causes PE2 to use Ethernet as its default VC type), PE2 can autosense and negotiate the interworking type and select VLAN as the VC type.

The table below summarizes shows the AC types, interworking options, and VC types after negotiation.

Table 13: Negotiating Ethernet and VLAN Interworking Types

PE1 AC Type	Interworking Option	PE2 AC Type	Interworking Option	VC Type after Negotiation
Ethernet	none	Ethernet	none	Ethernet
Vlan	none	Ethernet	none	Ethernet
Ethernet	none	Vlan	none	Ethernet
Vlan	none	Vlan	none	Ethernet
Ethernet	Vlan	Ethernet	none	Incompatible
Vlan	Vlan	Ethernet	none	Incompatible
Ethernet	Vlan	Vlan	none	Vlan
Vlan	Vlan	Vlan	none	Vlan
Ethernet	none	Ethernet	Vlan	Incompatible
Vlan	none	Ethernet	Vlan	Vlan
Ethernet	none	Vlan	Vlan	Incompatible

PE1 AC Type	Interworking Option	PE2 AC Type	Interworking Option	VC Type after Negotiation
Vlan	none	Vlan	Vlan	Vlan
Ethernet	Vlan	Ethernet	Vlan	Vlan
Vlan	Vlan	Ethernet	Vlan	Vlan
Ethernet	Vlan	Vlan	Vlan	Vlan
Vlan	Vlan	Vlan	Vlan	Vlan

Frame Relay Interworking Restrictions

The following restrictions apply to Frame Relay interworking:

- The attachment circuit maximum transmission unit (MTU) sizes must match when you connect them over MPLS. By default, the MTU size associated with a Frame Relay DLCI is the interface MTU. This may cause problems, for example, when connecting some DLCIs on a PoS interface (with a default MTU of 4470 bytes) to Ethernet or VLAN (with a default MTU of 1500 bytes) and other DLCIs on the same PoS interface to ATM (with a default MTU of 4470 bytes). To avoid reducing all the interface MTUs to the lowest common denominator (1500 bytes in this case), you can specify the MTU for individual DLCIs using the **mtu** command.
- Only DLCI mode is supported. Port mode is not supported.
- Configure Frame Relay switching to use DCE or Network-to-Network Interface (NNI). DTE mode does not report status in the Local Management Interface (LMI) process. If a Frame Relay over MPLS circuit goes down and the PE router is in DTE mode, the CE router is never informed of the disabled circuit. You must configure the **frame-relay switching** command in global configuration mode in order to configure DCE or NNI.
- Frame Relay policing is non-distributed on the Cisco 7500 series routers. If you enable Frame Relay policing, traffic is sent to the route switch processor for processing.
- Inverse ARP is not supported with IP interworking. CE routers must use point-to-point subinterfaces or static maps.
- The PE router automatically supports translation of both the Cisco encapsulations and the Internet Engineering Task Force (IETF) encapsulations that come from the CE, but translates only to IETF when sending to the CE router. This is not a problem for the Cisco CE router, because it can handle IETF encapsulation on receipt even if it is configured to send Cisco encapsulation.
- With Ethernet interworking, the following translations are supported:
  - Ethernet without LAN FCS (0300800080C20007 or 6558)
  - Spanning tree (0300800080C2000E or 4242)

All other translations are dropped.

- With IP interworking, the IPv4 (03CC or 0800) translation is supported. All other translations are dropped.
- PVC status signaling works the same way as in like-to-like case. The PE router reports the PVC status to the CE router, based on the availability of the pseudowire. PVC status detected by the PE router will also be reflected into the pseudowire. LMI to OAM interworking is supported when you connect Frame Relay to ATM.

PPP Interworking Restrictions

The following restrictions apply to PPP interworking:

- There must be a one-to-one relationship between a PPP session and the pseudowire. Multiplexing of multiple PPP sessions over the pseudowire is not supported.
- There must be a one-to-one relationship between a PPP session and a Frame Relay DLCI. Each Frame Relay PVC must have only one PPP session.
- Only IP (IPv4 (0021) interworking is supported. Link Control Protocol (LCP) packets and Internet Protocol Control Protocol (IPCP) packets are terminated at the PE router. Everything else is dropped.
- Proxy IPCP is automatically enabled on the PE router when IP interworking is configured on the pseudowire.
- By default, the PE router assumes that the CE router knows the remote CE router's IP address.
- Password Authentication Protocol (PAP) and Challenge-Handshake Authentication Protocol (CHAP) authentication are supported.

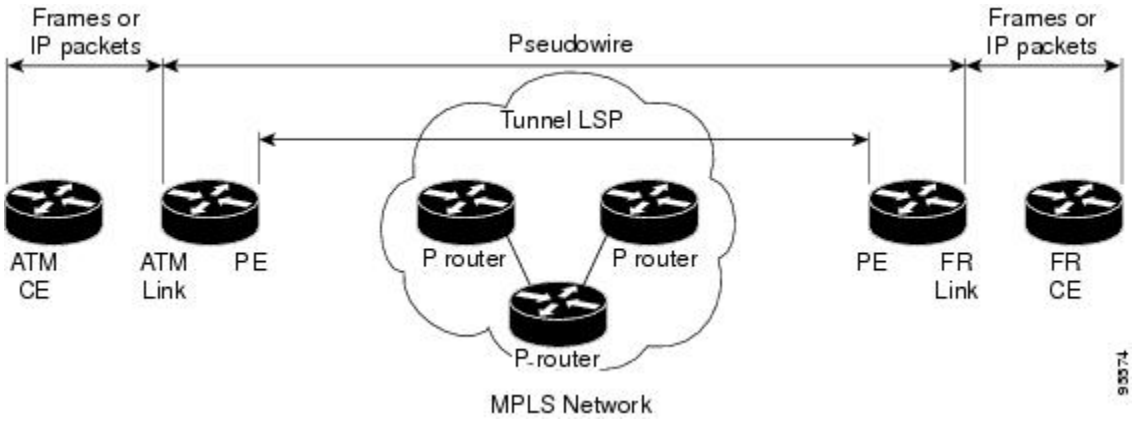
Information About L2VPN Interworking

Overview of L2VPN Interworking

Layer 2 transport over MPLS and IP already exists for like-to-like attachment circuits, such as Ethernet-to-Ethernet or PPP-to-PPP. L2VPN Interworking builds on this functionality by allowing disparate attachment circuits to be connected. An interworking function facilitates the translation between the different

Layer 2 encapsulations. The figure below is an example of Layer 2 interworking, where ATM and Frame Relay packets travel over the MPLS cloud.

Figure 4: ATM to Frame Relay Interworking Example



The L2VPN Interworking feature supports Ethernet, 802.1Q (VLAN), Frame Relay, ATM AAL5, and PPP attachment circuits over MPLS and L2TPv3. The features and restrictions for like-to-like functionality also apply to L2VPN Interworking.

L2VPN Interworking Modes

L2VPN Interworking works in either Ethernet ("bridged") mode, IP ("routed"), or Ethernet VLAN mode. You specify the mode by issuing the **interworking** {**ethernet** | **ip** |**vlan**} command in pseudowire-class configuration mode.

Ethernet (Bridged) Interworking

The **ethernet** keyword causes Ethernet frames to be extracted from the attachment circuit and sent over the pseudowire. Ethernet end-to-end transmission is assumed. Attachment circuit frames that are not Ethernet are dropped. In the case of VLAN, the VLAN tag is removed, leaving an untagged Ethernet frame.

Ethernet Interworking is also called bridged interworking. Ethernet frames are bridged across the pseudowire. The CE routers could be natively bridging Ethernet or could be routing using a bridged encapsulation model, such as Bridge Virtual Interface (BVI) or RBE. The PE routers operate in Ethernet like-to-like mode.

This mode is used to offer the following services:

- LAN services--An example is an enterprise that has several sites, where some sites have Ethernet connectivity to the service provider (SP) network and others have ATM connectivity. The enterprise wants LAN connectivity to all its sites. In this case, traffic from the Ethernet or VLAN of one site can be sent through the IP/MPLS network and encapsulated as bridged traffic over an ATM VC of another site.
- Connectivity services--An example is an enterprise that has different sites that are running an Internal Gateway Protocol (IGP) routing protocol, which has incompatible procedures on broadcast and nonbroadcast links. The enterprise has several sites that are running an IGP, such as Open Shortest Path First (OSPF) or Intermediate System to Intermediate System (IS-IS), between the sites. In this scenario,

some of the procedures (such as route advertisement or designated router) depend on the underlying Layer 2 protocol and are different for a point-to-point ATM connection versus a broadcast Ethernet connection. Therefore, the bridged encapsulation over ATM can be used to achieve homogenous Ethernet connectivity between the CE routers running the IGP.

IP (Routed) Interworking

The **ip** keyword causes IP packets to be extracted from the attachment circuit and sent over the pseudowire. Attachment circuit frames that do not contain IPv4 packets are dropped.

IP Interworking is also called routed interworking. The CE routers encapsulate IP on the link between the CE and PE routers. A new VC type is used to signal the IP pseudowire in MPLS and L2TPv3. Translation between the Layer 2 and IP encapsulations across the pseudowire is required. Special consideration needs to be given to address resolution and routing protocol operation, because these are handled differently on different Layer 2 encapsulations.

This mode is used to provide IP connectivity between sites, regardless of the Layer 2 connectivity to these sites. It is different from a Layer 3 VPN because it is point-to-point in nature and the service provider does not maintain any customer routing information.

Address resolution is encapsulation dependent:

- Ethernet uses ARP
- Frame Relay and ATM use Inverse ARP
- PPP uses IPCP

Therefore, address resolution must be terminated on the PE router. End-to-end address resolution is not supported. Routing protocols operate differently over broadcast and point-to-point media. For Ethernet, the CE routers must either use static routing or configure the routing protocols to treat the Ethernet side as a point-to-point network.

VLAN Interworking

The **vlan** keyword allows the VLAN ID to be included as part of the Ethernet frame. In Cisco IOS Release 12.2(52)SE, you can configure Catalyst 3750 Metro switches to use Ethernet VLAN for Ethernet (bridged) interworking. You can specify the Ethernet VLAN (type 4) by issuing the **interworking vlan** command in pseudowire-class configuration mode. This allows the VLAN ID to be included as part of the Ethernet frame. In releases previous to Cisco IOS Release 12.2(52)SE, the only way to achieve VLAN encapsulation is to ensure the CE router is connected to the PE router through an Ethernet VLAN interface/subinterface.

Before switching from Ethernet or IP interworking to Ethernet VLAN (type 4) interworking, ensure that you use the **clear mpls ldp neighbor** command in privileged EXEC mode to forcibly reset label distribution protocol (LDP) sessions. The **clear mpls ldp neighbor** command terminates the specified LDP sessions, which enables a renegotiation of the virtual circuit (VC) parameters. The LDP sessions should be reestablished if the LDP configuration remains unchanged.

You can clear an LDP session for an interface-specific label space of an LSR by issuing the **no mpls ip** command and then the **mpls ip** command on the interface associated with the LDP session.

The following example resets an LDP session:

```
Device# clear mpls ldp neighbor 10.0.0.10
```

To verify the results of the **clear mpls ldp neighbor** command, use the **show mpls ldp neighbor** command. Notice the value in the "Up time" field.

```
Device# show mpls ldp neighbor 10.0.0.10

 Peer LDP Ident: 10.0.0.10:0; Local LDP Ident 10.13.13.13:0
   TCP connection: 10.0.0.10.646 - 10.13.13.13.15093
   State: Oper; Msgs sent/rcvd: 142/138; Downstream
   Up time: 02:16:28
   LDP discovery sources:
    Serial1/0, Src IP addr: 10.0.0.2
   Addresses bound to peer LDP Ident:
    10.0.0.129      10.0.0.10     10.0.0.2        10.1.0.5
    10.7.0.1
```

L2VPN Interworking Support Matrix

The supported L2VPN Interworking features are listed in the table below.

Table 14: L2VPN Interworking Supported Features

Feature	MPLS or L2TPv3 Support	IP or Ethernet Support
Ethernet/VLAN to ATM AAL5	MPLS L2TPv3 (12000 series only)	IP Ethernet
Ethernet/VLAN to Frame Relay	MPLS L2TPv3	IP Ethernet
Ethernet/VLAN to PPP	MPLS	IP
Ethernet to VLAN	MPLS L2TPv3	IP Ethernet[1]
L2VPN Interworking: VLAN Enable/Disable Option for AToM	MPLS	Ethernet VLAN
Frame Relay to ATM AAL5	MPLS L2TPv3 (12000 series only)	IP
Frame Relay to Ethernet or VLAN	MPLS L2TPv3	IP Ethernet
Frame Relay to PPP	MPLS L2TPv3	IP
Note : On the Cisco 12000 series Internet router: • Ethernet (bridged) interworking is not supported for L2TPv3. • IP (routed) interworking is not supported in an L2TPv3 pseudowire configured for data sequencing (using the **sequencing** command).		

1 With the L2VPN Interworking: VLAN Enable/Disable Option for AToM feature, VLAN interworking can also be supported. For more information, see the "VLAN Interworking" section on page 14 .

Static IP Addresses for L2VPN Interworking for PPP

If the PE router needs to perform address resolution with the local CE router for PPP, you can configure the remote CE router's IP address on the PE router. Issue the **ppp ipcp address proxy** command with the remote CE router's IP address on the PE router's xconnect PPP interface. The following example shows a sample configuration:

```
pseudowire-class ip-interworking
 encapsulation mpls
 interworking ip
interface Serial2/0
 encapsulation ppp
 xconnect 10.0.0.2 200 pw-class ip-interworking
 ppp ipcp address proxy 10.65.32.14
```

You can also configure the remote CE router's IP address on the local CE router with the **peer default ip address** command if the local CE router performs address resolution.

How to Configure L2VPN Interworking

Configuring L2VPN Interworking

L2VPN Interworking allows you to connect disparate attachment circuits. Configuring the L2VPN Interworking feature requires that you add the **interworking** command to the list of commands that make up the pseudowire. The steps for configuring the pseudowire for L2VPN Interworking are included in this section. You use the **interworking**command as part of the overall AToM or L2TPv3 configuration. For specific instructions on configuring AToM or L2TPv3, see the following documents:

- Layer 2 Tunnel Protocol Version 3
- Any Transport over MPLS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hw-module slot** *slot-number* **np mode feature**
4. **pseudowire-class** *name*
5. **encapsulation** {**mpls** | **l2tpv3**}
6. **interworking** {**ethernet** | **ip**} | **vlan**}

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	**hw-module slot** *slot-number* **np mode feature** **Example:** `Router(config)# hw-module slot 3 np mode feature`	(Optional) Enables L2VPN Interworking functionality on the Cisco 12000 series router. **Note** Enter this command only on a Cisco 12000 series Internet router if you use L2TPv3 for L2VPN Interworking on an ISE (Engine 3) or Engine 5 interface. In this case, you must first enable the L2VPN feature bundle on the line card by entering the **hw-module slot** *slot-number* **np mode feature** command.
Step 4	**pseudowire-class** *name* **Example:** `Router(config)# pseudowire-class class1`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 5	**encapsulation** {**mpls** | **l2tpv3**} **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation, which is either **mpls** or **l2tpv3**.
Step 6	**interworking** {**ethernet** | **ip**} | **vlan**} **Example:** `Router(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it. **Note** On the Cisco 12000 series Internet router, Ethernet (bridged) interworking is not supported for L2TPv3. After you configure the L2TPv3 tunnel encapsulation for the pseudowire using the **encapsulation l2tpv3**command, you cannot enter the **interworking ethernet** command.

Verifying the L2VPN Interworking Configuration

To verify the L2VPN Interworking configuration, you can use the following commands.

SUMMARY STEPS

1. **enable**
2. **show l2tun session all (L2TPv3 only)**
3. **show arp**
4. **ping**
5. **show l2tun session interworking (L2TPv3 only)**
6. **show mpls l2transport vc detail (AToM only)**

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **show l2tun session all (L2TPv3 only)**
For L2TPv3, you can verify the L2VPN Interworking configuration using the **show l2tun session all** command on the PE routers.

In the following example, the interworking type is shown in bold.

PE1	PE2

PE1	PE2
Router# **show l2tun session all** Session Information Total tunnels 1 sessions 1 Session id 15736 is up, tunnel id 35411 Call serial number is 4035100045 Remote tunnel name is PE2 Internet address is 10.9.9.9 Session is L2TP signalled Session state is established, time since change 1d22h 16 Packets sent, 16 received 1518 Bytes sent, 1230 received Receive packets dropped: out-of-order: 0 total: 0 Send packets dropped: exceeded session MTU: 0 total: 0 Session vcid is 123 Session Layer 2 circuit, type is Ethernet, name is FastEthernet1/1/0 Circuit state is UP Remote session id is 26570, remote tunnel id 46882 DF bit off, ToS reflect disabled, ToS value 0, TTL value 255 No session cookie information available	Router# **show l2tun session all** Session Information Total tunnels 1 sessions 1 Session id 26570 is up, tunnel id 46882 Call serial number is 4035100045 Remote tunnel name is PE1 Internet address is 10.8.8.8 Session is L2TP signalled Session state is established, time since change 1d22h 16 Packets sent, 16 received 1230 Bytes sent, 1230 received Receive packets dropped: out-of-order: 0 total: 0 Send packets dropped: exceeded session MTU: 0 total: 0 Session vcid is 123 Session Layer 2 circuit, type is Ethernet Vlan, name is FastEthernet2/0.1:10 Circuit state is UP, **interworking type is Ethernet** Remote session id is 15736, remote tunnel id 35411 DF bit off, ToS reflect disabled, ToS value 0, TTL value 255 No session cookie information available

PE1	PE2
FS cached header information: encap size = 24 bytes 00000000 00000000 00000000 00000000 00000000 00000000 Sequencing is off	FS cached header information: encap size = 24 bytes 00000000 00000000 00000000 00000000 00000000 00000000 Sequencing is off

Step 3 **show arp**

You can issue the **show arp** command between the CE routers to ensure that data is being sent:

Example:

```
Router# show arp
Protocol   Address       Age (min)   Hardware Addr    Type    Interface
Internet   10.1.1.5           134    0005.0032.0854   ARPA    FastEthernet0/0
Internet   10.1.1.7             -    0005.0032.0000   ARPA    FastEthernet0/0
```

Step 4 **ping**

You can issue the **ping** command between the CE routers to ensure that data is being sent:

Example:

```
Router# ping 10.1.1.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Step 5 **show l2tun session interworking (L2TPv3 only)**

For L2TPv3, you can verify that the interworking type is correctly set using the **show l2tun session interworking** command. Enter the command on the PE routers that are performing the interworking translation.

- In Example 1, the PE router performs the raw Ethernet translation. The command output displays the interworking type with a dash (-).
- In Example 2, the PE router performs the Ethernet VLAN translation. The command output displays the interworking type as ETH.

Command Output for Raw Ethernet Translation

Example:

```
Router# show l2tun session interworking
Session Information Total tunnels 1 sessions 1
LocID      TunID      Peer-address    Type IWrk Username, Intf/Vcid, Circuit
15736      35411      10.9.9.9         ETH   -   123,      Fa1/1/0
```

Command Output for Ethernet VLAN Translation

Example:

```
Router# show l2tun session interworking
Session Information Total tunnels 1 sessions 1
LocID      TunID      Peer-address    Type IWrk Username, Intf/Vcid, Circuit
26570      46882      10.8.8.8         VLAN ETH  123,      Fa2/0.1:10
```

Step 6 **show mpls l2transport vc detail (AToM only)**

You can verify the AToM configuration by using the **show mpls l2transport vc detail** command. In the following example, the interworking type is shown in bold.

PE1	PE2
Router# **show mpls l2transport vc detail** Local interface: Fa1/1/0 up, line protocol up, Ethernet up Destination address: 10.9.9.9, VC ID: 123, VC status: up Preferred path: not configured Default path: active Tunnel label: 17, next hop 10.1.1.3 Output interface: Fa4/0/0, imposed label stack {17 20} Create time: 01:43:50, last status change time: 01:43:33 Signaling protocol: LDP, peer 10.9.9.9:0 up MPLS VC labels: local 16, remote 20 Group ID: local 0, remote 0 MTU: local 1500, remote 1500 Remote interface description: Sequencing: receive disabled, send disabled VC statistics: packet totals: receive 15, send 4184 byte totals: receive 1830, send 309248 packet drops: receive 0, send 0	Router# **show mpls l2transport vc detail** Local interface: Fa2/0.3 up, line protocol up, Eth VLAN 10 up MPLS VC type is Ethernet, **interworking type is Ethernet** Destination address: 10.8.8.8, VC ID: 123, VC status: up Preferred path: not configured Default path: active Tunnel label: 16, next hop 10.1.1.3 Output interface: Fa6/0, imposed label stack {16 16} Create time: 00:00:26, last status change time: 00:00:06 Signaling protocol: LDP, peer 10.8.8.8:0 up MPLS VC labels: local 20, remote 16 Group ID: local 0, remote 0 MTU: local 1500, remote 1500 Remote interface description: Sequencing: receive disabled, send disabled VC statistics: packet totals: receive 5, send 0 byte totals: receive 340, send 0 packet drops: receive 0, send 0

Configuring L2VPN Interworking: VLAN Enable-Disable Option for AToM

You can specify the Ethernet VLAN (type 4) by issuing the **interworking vlan** command in pseudowire-class configuration mode. This allows the VLAN ID to be included as part of the Ethernet frame. In releases previous to Cisco IOS Release 12.2(52)SE and Cisco IOS Release 12.2(33)SRE, the only way to achieve VLAN encapsulation is to ensure the CE router is connected to the PE router through an Ethernet link.

Before switching from Ethernet or IP interworking to Ethernet VLAN (type 4) interworking, ensure that you use the **clear mpls ldp neighbor** command in privileged EXEC mode to forcibly reset label distribution protocol (LDP) sessions. The **clear mpls ldp neighbor** command terminates the specified LDP sessions, which enables a renegotiation of the virtual circuit (VC) parameters. The LDP sessions should be reestablished if the LDP configuration remains unchanged.

You can clear an LDP session for an interface-specific label space of an LSR by issuing the **no mpls ip** command and then the **mpls ip** command on the interface associated with the LDP session.

The following example resets an LDP session:

```
Device# clear mpls ldp neighbor 10.0.0.10
```

To verify the results of the **clear mpls ldp neighbor** command, use the **show mpls ldp neighbor** command. Notice the value in the "Up time" field.

```
Device# show mpls ldp neighbor 10.0.0.10

 Peer LDP Ident: 10.0.0.10:0; Local LDP Ident 10.13.13.13:0
   TCP connection: 10.0.0.10.646 - 10.13.13.13.15093
   State: Oper; Msgs sent/rcvd: 142/138; Downstream
   Up time: 02:16:28
   LDP discovery sources:
    Serial1/0, Src IP addr: 10.0.0.2
   Addresses bound to peer LDP Ident:
    10.0.0.129       10.0.0.10     10.0.0.2        10.1.0.5
    10.7.0.1
```

Before You Begin

For complete instructions on configuring AToM, see "Any Transport over MPLS".

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *name*
4. **encapsulation** {**mpls** | **l2tpv3**}
5. **interworking** {**ethernet** | **ip**| **vlan**}
6. **end**
7. **show mpls l2transport vc** [**vcid** *vc-id* | **vcid** *vc-id-min vc-id-max*] [**interface** *type number* [*local-circuit-id*]] [**destination** *ip-address* | *name*] [**detail**]

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** *name* **Example:** `Router(config)# pseudowire-class class1`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 4	**encapsulation** {**mpls** | **l2tpv3**} **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation, which is either **mpls** or **l2tpv3**. • For the L2VPN Internetworking: VLAN Enable/Disable option for AToM feature, only MPLS encapsulation is supported.
Step 5	**interworking** {**ethernet** | **ip**| **vlan**} **Example:** `Router(config-pw)# interworking vlan`	Specifies the type of pseudowire and the type of traffic that can flow across it. • For the L2VPN Internetworking: VLAN Enable/Disable option for AToM feature, specify the **vlan** keyword.
Step 7	**end** **Example:** `Router(config-pw)# end`	Exits pseudowire class configuration mode and enters privileged EXEC mode.

Step 7	**show mpls l2transport vc** [**vcid** *vc-id* | **vcid** *vc-id-min vc-id-max*] [**interface** *type number* [*local-circuit-id*]] [**destination** *ip-address* | *name*] [**detail**]
	
	Example:
	`Router#`
	`show mpls l2transport vc detail`
		Displays information about AToM VCs.

Examples

When the pseudowire on an interface is different from the VC type, the interworking type is displayed in the **show mpls l2transport vc detail** command output. In the following example, the pseudowire is configured on an Ethernet port and VLAN interworking is configured in the pseudowire class. The relevant output is shown in bold:

```
PE1# show mpls l2 vc 34 detail
Local interface: Et0/1 up, line protocol up, Ethernet up
  MPLS VC type is Ethernet, interworking type is Eth VLAN
  Destination address: 10.1.1.2, VC ID: 34, VC status: down
    Output interface: if-?(0), imposed label stack {}
    Preferred path: not configured
    Default path: no route
    No adjacency
  Create time: 00:00:13, last status change time: 00:00:13
  Signaling protocol: LDP, peer unknown
    Targeted Hello: 10.1.1.1(LDP Id) -> 10.1.1.2
    Status TLV support (local/remote)   : enabled/None (no remote binding)
      LDP route watch                   : enabled
      Label/status state machine        : local standby, AC-ready, LnuRnd
      Last local dataplane   status rcvd: No fault
      Last local SSS circuit status rcvd: No fault
      Last local SSS circuit status sent: Not sent
      Last local  LDP TLV    status sent: None
      Last remote LDP TLV    status rcvd: None (no remote binding)
      Last remote LDP ADJ    status rcvd: None (no remote binding)
    MPLS VC labels: local 2003, remote unassigned
    Group ID: local 0, remote unknown
    MTU: local 1500, remote unknown
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 0, send 0
    byte totals:   receive 0, send 0
    packet drops:  receive 0, seq error 0, send 0
```

Configuration Examples for L2VPN Interworking

Ethernet to VLAN over L2TPV3 (Bridged) Example

The following example shows the configuration of Ethernet to VLAN over L2TPv3:

PE1	PE2
ip cef ! l2tp-class interworking-class authentication hostname PE1 password 0 lab ! pseudowire-class inter-ether-vlan encapsulation l2tpv3 interworking ethernet protocol l2tpv3 interworking-class ip local interface Loopback0 ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 ! interface FastEthernet1/0 xconnect 10.9.9.9 1 pw-class inter-ether-vlan	ip cef ! l2tp-class interworking-class authentication hostname PE2 password 0 lab ! pseudowire-class inter-ether-vlan encapsulation l2tpv3 interworking ethernet protocol l2tpv3 interworking-class ip local interface Loopback0 ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 ! interface FastEthernet0/0 no ip address ! interface FastEthernet0/0.3 encapsulation dot1Q 10 xconnect 10.8.8.8 1 pw-class inter-ether-vlan

Ethernet to VLAN over AToM (Bridged) Example

The following example shows the configuration of Ethernet to VLAN over AToM:

PE1	PE2
ip cef ! mpls label protocol ldp mpls ldp router-id Loopback0 force ! pseudowire-class atom-eth-iw encapsulation mpls interworking ethernet ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 ! interface FastEthernet1/0.1 encapsulation dot1q 100 xconnect 10.9.9.9 123 pw-class atom-eth-iw	ip cef ! mpls label protocol ldp mpls ldp router-id Loopback0 force ! pseudowire-class atom encapsulation mpls ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 ! interface FastEthernet0/0 no ip address ! interface FastEthernet1/0 xconnect 10.9.9.9 123 pw-class atom

Frame Relay to VLAN over L2TPV3 (Routed) Example

The following example shows the configuration of Frame Relay to VLAN over L2TPv3:

PE1	PE2

configure terminal
ip cef
frame-relay switching
!
!
interface loopback 0
ip address 10.8.8.8 255.255.255.255
no shutdown
!
pseudowire-class ip
 encapsulation l2tpv3
 interworking ip
 ip local interface loopback0
!
interface POS1/0
encapsulation frame-relay
clock source internal
logging event dlci-status-change
no shutdown
no fair-queue
!
connect fr-vlan POS1/0 206 l2transport
 xconnect 10.9.9.9 6 pw-class ip
!
router ospf 10
 network 10.0.0.2 0.0.0.0 area 0
 network 10.8.8.8 0.0.0.0 area 0
	configure terminal
	ip routing
	ip cef
	frame-relay switching
	!
	interface loopback 0
	ip address 10.9.9.9 255.255.255.255
	no shutdown
	!
	pseudowire-class ip
	 encapsulation l2tpv3
	 interworking ip
	 ip local interface loopback0
	!
	interface FastEthernet1/0/1
	  speed 10
	  no shutdown
	!
	interface FastEthernet1/0/1.6
	encapsulation dot1Q 6
	xconnect 10.8.8.8 6 pw-class ip
	no shutdown
	!
	router ospf 10
	 network 10.0.0.2 0.0.0.0 area 0
	 network 10.9.9.9 0.0.0.0 area 0

Frame Relay to VLAN over AToM (Routed) Example

The following example shows the configuration of Frame Relay to VLAN over AToM:

PE1	PE2
configure terminal ip cef frame-relay switching ! mpls label protocol ldp mpls ldp router-id loopback0 mpls ip ! pseudowire-class atom encapsulation mpls interworking ip ! interface loopback 0 ip address 10.8.8.8 255.255.255.255 no shutdown ! connect fr-vlan POS1/0 206 l2transport xconnect 10.9.9.9 6 pw-class atom	configure terminal ip routing ip cef frame-relay switching ! mpls label protocol ldp mpls ldp router-id loopback0 mpls ip ! pseudowire-class atom encapsulation mpls interworking ip ! interface loopback 0 ip address 10.9.9.9 255.255.255.255 no shutdown ! interface FastEthernet1/0/1.6 encapsulation dot1Q 6 xconnect 10.8.8.8 6 pw-class atom no shutdown

Frame Relay to ATM AAL5 over AToM (Routed) Example

Note Frame Relay to ATM AAL5 is available only with AToM in IP mode.

The following example shows the configuration of Frame Relay to ATM AAL5 over AToM:

PE1	PE2
ip cef frame-relay switching mpls ip mpls label protocol ldp mpls ldp router-id loopback0 force pseudowire-class fratmip encapsulation mpls interworking ip interface Loopback0 ip address 10.33.33.33 255.255.255.255 interface serial 2/0 encapsulation frame-relay ietf frame-relay intf-type dce connect fr-eth serial 2/0 100 l2transport xconnect 10.22.22.22 333 pw-class fratmip interface POS1/0 ip address 10.1.7.3 255.255.255.0 crc 32 clock source internal mpls ip mpls label protocol ldp router ospf 10 passive-interface Loopback0 network 10.33.33.33 0.0.0.0 area 10 network 10.1.7.0 0.0.0.255 area 10	ip cef mpls ip mpls label protocol ldp mpls ldp router-id loopback0 force pseudowire-class fratmip encapsulation mpls interworking ip interface Loopback0 ip address 10.22.22.22 255.255.255.255 interface ATM 2/0 pvc 0/203 l2transport encapsulation aa5snap xconnect 10.33.33.33 333 pw-class fratmip interface POS1/0 ip address 10.1.1.2 255.255.255.0 crc 32 clock source internal mpls ip mpls label protocol ldp router ospf 10 passive-interface Loopback0 network 10.22.22.22 0.0.0.0 area 10 network 10.1.1.0 0.0.0.255 area 10

VLAN to ATM AAL5 over AToM (Bridged) Example

The following example shows the configuration of VLAN to ATM AAL5 over AToM:

PE1	PE2
ip cef ! mpls ip mpls label protocol ldp mpls ldp router-id Loopback0 ! pseudowire-class inter-ether encapsulation mpls interworking ethernet ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 ! interface ATM1/0.1 point-to-point pvc 0/100 l2transport encapsulation aal5snap xconnect 10.9.9.9 123 pw-class inter-ether ! interface FastEthernet1/0 xconnect 10.9.9.9 1 pw-class inter-ether ! router ospf 10 log-adjacency-changes network 10.8.8.8 0.0.0.0 area 0 network 10.1.1.1 0.0.0.0 area 0	ip cef ! mpls ip mpls label protocol ldp mpls ldp router-id Loopback0 ! pseudowire-class inter-ether encapsulation mpls interworking ethernet ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 ! interface FastEthernet0/0 no ip address ! interface FastEthernet0/0.1 encapsulation dot1Q 10 xconnect 10.8.8.8 123 pw-class inter-ether ! router ospf 10 log-adjacency-changes network 10.9.9.9 0.0.0.0 area 0 network 10.1.1.2 0.0.0.0 area 0

Frame Relay to PPP over L2TPv3 (Routed) Example

The following example shows the configuration of Frame Relay to PPP over L2TPv3:

PE1	PE2
ip cef ip routing ! ! ! pseudowire-class ppp-fr encapsulation l2tpv3 interworking ip ip local interface Loopback0 ! interface Loopback0 ip address 10.1.1.1 255.255.255.255 ! interface FastEthernet1/0/0 ip address 10.16.1.1 255.255.255.0 ! interface Serial3/0/0 no ip address encapsulation ppp ppp authentication chap ! ip route 10.0.0.0 255.0.0.0 10.16.1.2 ! xconnect 10.2.2.2 1 pw-class ppp-fr ppp ipcp address proxy 10.65.32.14	ip cef ip routing ! frame-relay switching ! pseudowire-class ppp-fr encapsulation l2tpv3 interworking ip ip local interface Loopback0 ! interface Loopback0 ip address 10.2.2.2 255.255.255.255 ! interface FastEthernet1/0/0 ip address 10.16.2.1 255.255.255.0 ! interface Serial3/0/0 no ip address encapsulation frame-relay frame-relay intf-type dce ! ip route 10.0.0.0 255.0.0.0 10.16.2.2 ! connect ppp-fr Serial3/0/0 100 l2transport xconnect 10.1.1.1 100 pw-class ppp-fr

Frame Relay to PPP over AToM (Routed) Example

The following example shows the configuration of Frame Relay to PPP over AToM:

PE1	PE2
ip cef ip routing mpls label protocol ldp mpls ldp router-id loopback0 force ! ! ! pseudowire-class ppp-fr encapsulation mpls interworking ip ip local interface Loopback0 ! interface Loopback0 ip address 10.1.1.1 255.255.255.255 ! interface FastEthernet1/0/0 ip address 10.16.1.1 255.255.255.0 mpls ip label protocol ldp ! interface Serial3/0/0 no ip address encapsulation ppp ppp authentication chap xconnect 10.2.2.2 1 pw-class ppp-fr ppp ipcp address proxy 10.65.32.14 ! ip route 10.0.0.0 255.0.0.0 10.16.1.2	ip cef ip routing mpls label protocol ldp mpls ldp router-id loopback0 force ! frame-relay switching ! pseudowire-class ppp-fr encapsulation mpls interworking ip ip local interface Loopback0 ! interface Loopback0 ip address 10.2.2.2 255.255.255.255 ! interface FastEthernet1/0/0 ip address 10.16.2.1 255.255.255.0 mpls ip mpls label protocol ldp ! interface Serial3/0/0 no ip address encapsulation frame-relay frame-relay intf-type dce ! ip route 10.0.0.0 255.0.0.0 10.16.2.2 ! connect ppp-fr Serial3/0/0 100 l2transport xconnect 10.1.1.1 100 pw-class ppp-fr

Ethernet VLAN to PPP over AToM (Routed) Example

The following example shows the configuration of Ethernet VLAN to PPP over AToM:

PE1	PE2
configure terminal mpls label protocol ldp mpls ldp router-id Loopback0 mpls ip ! pseudowire-class ppp-ether encapsulation mpls interworking ip ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 no shutdown ! interface POS2/0/1 no ip address encapsulation ppp no peer default ip address ppp ipcp address proxy 10.10.10.1 xconnect 10.9.9.9 300 pw-class ppp-ether no shutdown	configure terminal mpls label protocol ldp mpls ldp router-id Loopback0 mpls ip ! pseudowire-class ppp-ether encapsulation mpls interworking ip ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 no shutdown ! interface vlan300 mtu 4470 no ip address xconnect 10.8.8.8 300 pw-class ppp-ether no shutdown ! interface GigabitEthernet6/2 switchport switchport trunk encapsulation dot1q switchport trunk allowed vlan 300 switchport mode trunk no shutdown

Additional References for L2VPN Interworking

Related Documents

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

Standards and RFCs

Standard/RFC	Title
draft-ietf-l2tpext-l2tp-base-03.txt	*Layer Two Tunneling Protocol (Version 3) 'L2TPv3'*
draft-martini-l2circuit-trans-mpls-09.txt	*Transport of Layer 2 Frames Over MPLS*
draft-ietf-pwe3-frame-relay-03.txt.	*Encapsulation Methods for Transport of Frame Relay over MPLS Networks*
draft-martini-l2circuit-encap-mpls-04.txt.	*Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks*
draft-ietf-pwe3-ethernet-encap-08.txt.	*Encapsulation Methods for Transport of Ethernet over MPLS Networks*
draft-ietf-pwe3-hdlc-ppp-encap-mpls-03.txt.	*Encapsulation Methods for Transport of PPP/HDLC over MPLS Networks*
draft-ietf-ppvpn-l2vpn-00.txt.	*An Architecture for L2VPNs*
RFC 4618	Encapsulation Methods for Transport of PPP/High-Level Data Link Control (HDLC) over MPLS Networks

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 IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

Feature Information for L2VPN Interworking

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

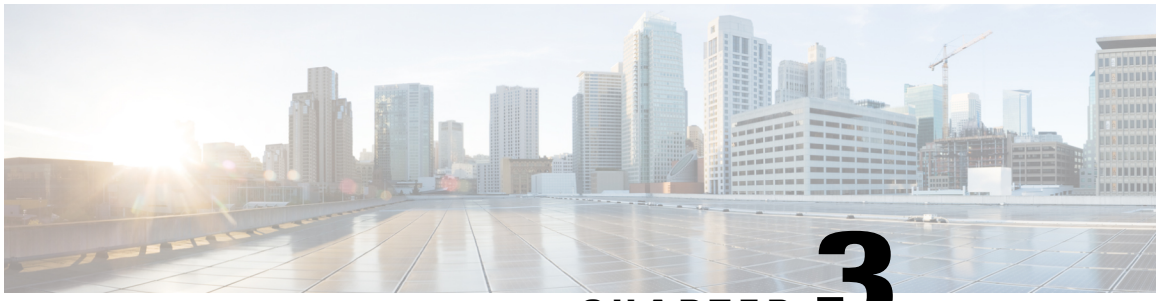
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 15: Feature Information for L2VPN Interworking

Feature Name	Releases	Feature Information
L2VPN Interworking	12.0(26)S 12.0(30)S 12.0(32)S 12.0(32)SY 12.2(33)SRA 12.4(11)T 12.2(33)SXH 12.2(33)SRD 12.2(52)SE 12.2(33)SRE	

Feature Name	Releases	Feature Information
		This feature allows disparate attachment circuits to be connected. An interworking function facilitates the translation between the different Layer 2 encapsulations. This feature was introduced in Cisco IOS Release 12.0(26)S. In Cisco IOS Release 12.0(30)S, support was added for Cisco 12000 series Internet routers. In Cisco IOS Release 12.0(32)S, support was added on Engine 5 line cards (SIP-401, SIP-501, SIP-600, and SIP-601) in Cisco 12000 series routers for the following four transport types: • Ethernet/VLAN to Frame Relay Interworking • Ethernet/VLAN to ATM AAL5 Interworking • Ethernet to VLAN Interworking • Frame Relay to ATM AAL5 Interworking On the Cisco 12000 series Internet router, support was added for IP Services Engine (ISE) and Engine 5 line cards that are configured for L2TPv3 tunneling. In Cisco IOS Release 12.2(33)SRA, support was added for the Cisco 7600 series routers. In Cisco IOS Release 12.4(11)T, support was added for the following transport types: • Ethernet to VLAN Interworking • Ethernet/VLAN to Frame Relay Interworking This feature was integrated into Cisco IOS Release 12.2(33)SXH.

Feature Name	Releases	Feature Information
		In Cisco IOS Release 12.2(33)SRD, support for routed and bridged interworking on SIP-400 was added for the Cisco 7600 series routers. In Cisco IOS Release 12.2(52)SE, the L2VPN Internetworking: VLAN Enable/Disable option for AToM feature was added for the Cisco 3750 Metro switch. In Cisco IOS Release 12.2(33)SRE, the L2VPN Internetworking: VLAN Enable/Disable option for AToM feature was added for the Cisco 7600 series router. The following commands were introduced or modified: **interworking**



CHAPTER 3

MPLS Pseudowire Status Signaling

The MPLS Pseudowire Status Signaling feature enables you to configure the router so it can send pseudowire status to a peer router, even when the attachment circuit is down. In releases prior to Cisco IOS 12.2(33)SRC, if the attachment circuit was down, the pseudowire status messages were not sent to the peer.

Finding Feature Information

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

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

Prerequisites for MPLS Pseudowire Status Signaling

- Before configuring this feature, make sure that both peer routers are capable of sending and receiving pseudowire status messages. Specifically, both routers should be running Cisco IOS Release 12.2(33)SRC and have the supported hardware installed.

Restrictions for MPLS Pseudowire Status Signaling

- Both peer routers must support the ability to send and receive pseudowire status messages in label advertisement and label notification messages. If both peer routers do not support pseudowire status messages, Cisco recommends that you disable the messages with the **no status** command.
- This feature is not integrated with Any Transport over MPLS (AToM) Virtual Circuit Connection Verification (VCCV).
- This feature is not integrated with Bidirectional Forwarding Detection (BFD).
- The standby and required switchover values from IETF draft-muley-pwe3-redundancy-02.txt are not supported.
- For a list of supported hardware for this feature, see the release notes for your platform.

Information About MPLS Pseudowire Status Signaling

How MPLS Pseudowire Status Signaling Works

In releases prior to Cisco IOS Release 12.2(33)SRC, the control plane for AToM does not have the ability to provide pseudowire status. Therefore, when an attachment circuit (AC) associated with a pseudowire is down (or is forced down as part of the Pseudowire Redundancy functionality), labels advertised to peers are withdrawn. In Cisco IOS Release 12.2(33)SRC, the MPLS Pseudowire Status Signaling feature enables the AC status to be sent to the peer through the Label Distribution Protocol.

The pseudowire status messages are sent in label advertisement and label notification messages if the peer also supports the MPLS Pseudowire Status Signaling feature. You can issue the **show mpls l2transport vc detail** command to show that both the local and remote routers support pseudowire status messages. The following example shows the line of output to look for:

```
Router# show mpls l2transport vc detail
.
.
.
status TLV support (local/remote): enabled/supported
```

When One Router Does Not Support MPLS Pseudowire Status Signaling

The peer routers must support the ability to send and receive pseudowire status messages in label advertisement and label notification messages. If one router does not support pseudowire status messages, Cisco recommends that you disable the messages with the **no status** command. This returns the router to label withdraw mode.

If the peer does not support the MPLS Pseudowire Status Signaling feature, the local router changes its mode of operation to label withdraw mode. You can issue the **show mpls l2transport vc detail** command to show

that the remote router does not support pseudowire status messages. The following example shows the line of output to look for:

```
Router# show mpls l2transport vc detail
.
.
.
status TLV support (local/remote): enabled/not supported
```

When you issue the following **debug mpls l2transport vc**commands, the messages show that the peer router does not supportthe MPLS Pseudowire Status Signaling feature and that the local router is changing to withdraw mode, as shown in bold in the following example:

Router# debug mpls l2transport vc event Router# **debug mpls l2transport vc status event** Router# **debug mpls l2transport vc status fsm** Router# **debug mpls l2transport vc ldp**

*Feb 26 13:41:40.707: AToM LDP [110.1.1.2]: Sending label withdraw msg *Feb 26 13:41:40.707: AToM LDP [110.1.1.2]: VC Type 5, mtu 1500 *Feb 26 13:41:40.707: AToM LDP [110.1.1.2]: VC ID 100, label 18 *Feb 26 13:41:40.707: AToM LDP [110.1.1.2]: Status 0x0000000A [PW Status NOT supported]

Status Messages Indicating That the Attachment Circuit Is Down

When the attachment circuit is down between the two routers, the output of the **show mpls l2transport vc detail** command shows the following status:

```
Router# show mpls l2transport vc detail
.
.
.
Last remote LDP TLV    status rcvd: AC DOWN(rx,tx faults)
```

The debug messages also indicate that the attachment circuit is down, as shown in bold in the command output:

Router# debug mpls l2transport vc event Router# **debug mpls l2transport vc status event** Router# **debug mpls l2transport vc status fsm** Router# **debug mpls l2transport vc ldp**

```
*Feb 26 11:51:42.427: AToM LDP [10.1.1.1]: Received notif msg, id 88
*Feb 26 11:51:42.427: AToM LDP [10.1.1.1]:   Status    0x00000007 [PW Status]
*Feb 26 11:51:42.427: AToM LDP [10.1.1.1]:   PW Status 0x00000006 [AC DOWN(rx,tx faults)]
```

Other pseudowire status messages include not-forwarding, pw-tx-fault, and pw-rx-fault.

Message Codes in the Pseudowire Status Messages

The **debug mpls l2transport vc**and the **show mpls l2transport vc detail** commands show output that contains message codes. For example:

```
    Label/status state machine: established, LruRru

AToM MGR [10.9.9.9, 100]: S:Evt local up, LndRru->LnuRru
```

The message codes (LruRru, LndRru, and LnuRru) indicate the status of the local and remote routers. You can use the following key to interpret the message codes:

L—local router

R—remote router

r or n—ready (r) or not ready (n)

u or d—up (u) or down (d) status

The output also includes other values:

D—Dataplane

S—Local shutdown

How to Configure MPLS Pseudowire Status Signaling

Enabling MPLS Pseudowire Status Signaling

Perform the following task to enable the router to send pseudowire status to a peer router even when the attachment circuit is down. If both routers do not support pseudowire status messages, then disable the messages with the **no status** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *name*
4. **status**
5. **encapsulation mpls**
6. **exit**
7. **exit**
8. **show mpls l2transport vc detail**

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	**pseudowire-class** *name* **Example:** `Router(config)# pseudowire-class atom`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 4	**status** **Example:** `Router(config-pw)# status`	(Optional) Enables the router to send pseudowire status messages to the peer router through label advertisement and label notification messages. **Note** By default, status messages are enabled. This step is included only in case status messages have been disabled. If you need to disable status messages because both peer routers do not support this functionality, enter the **no status**command.
Step 5	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 6	**exit** **Example:** `Router(config-pw)# exit`	Exits pseudowire class configuration mode.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 8	**show mpls l2transport vc detail** **Example:** `Router# show mpls l2transport vc detail`	Validates that pseudowire messages can be sent and received.

Configuration Examples for MPLS Pseudowire Status Signaling

MPLS Pseudowire Status Signaling Example

The following example configures the MPLS Pseudowire Status Signaling feature on two PE routers. By default, status messages are enabled. The **status** command is included in this example in case status messages have been disabled.

PE1

```
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
pseudowire-class atomstatus
 encapsulation mpls
 status
!
interface GigabitEthernet10/5
 xconnect 10.1.1.2 123 pw-class atomstatus
```

PE2

```
interface Loopback0
 ip address 10.1.1.2 255.255.255.255
!
pseudowire-class atomstatus
 encapsulation mpls
 status
!
interface GigabitEthernet3/3
 xconnect 10.1.1.1 123 pw-class atomstatus
```

Verifying That Both Routers Support Pseudowire Status Messages Example

You can issue the **show mpls l2transport vc detail** command to show that both the local and remote routers support pseudowire status messages. The following example shows the line of output to look for:

```
Router# show mpls l2transport vc detail

.

.

.

status TLV support (local/remote): enabled/supported
```

Additional References

The following sections provide references related to the MPLS Pseudowire Status Signaling feature.

Related Documents

Related Topic	Document Title
Any Transport over MPLS	Any Transport over MPLS
Virtual Private LAN Services	Virtual Private LAN Services on the Optical Services Modules

Standards

Standard	Title
draft-ietf-pwe3-control-protocol-15.txt	Pseudowire Setup and Maintenance Using LDP
draft-ietf-pwe3-iana-allocation-08.txt	IANA Allocations for Pseudo Wire Edge to Edge Emulation (PWE3)
draft-martini-pwe3-pw-switching-03.txt	Pseudo Wire Switching

MIBs

MIB	MIBs Link
Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	—

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

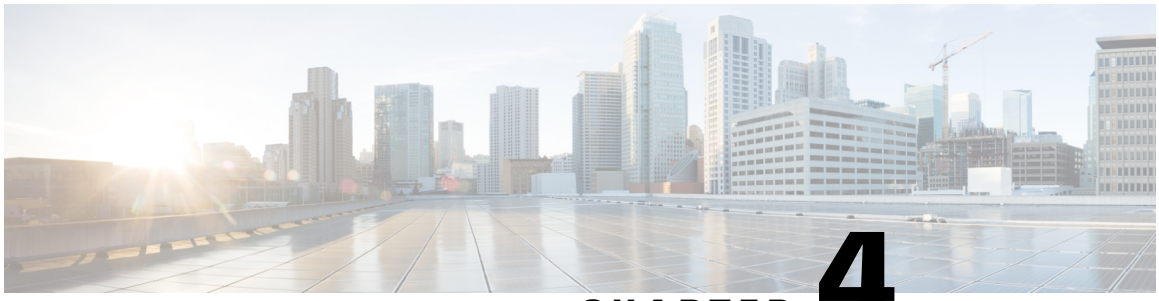
Feature Information for MPLS Pseudowire Status Signaling

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 16: Feature Information for MPLS Pseudowire Status Signaling

Feature Name	Releases	Feature Information
MPLS Pseudowire Status Signaling	12.2(33)SRC 12.2(50)SY	The MPLS Pseudowire Status Signaling feature enables you to configure the router so that it can send the pseudowire status to a peer router, even when the attachment circuit is down. The following commands were introduced or modified: **debug mpls l2transport vc**, **show mpls l2transport vc**, **status** (pseudowire class).



CHAPTER 4

L2VPN Pseudowire Redundancy

The L2VPN Pseudowire Redundancy feature lets you configure your network to detect a failure in the network and reroute the Layer 2 (L2) service to another endpoint that can continue to provide service. This feature provides the ability to recover from a failure either of the remote provider edge (PE) router or of the link between the PE and customer edge (CE) routers.

Finding Feature Information

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

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

Prerequisites for L2VPN Pseudowire Redundancy

- This feature module requires that you understand how to configure basic L2 virtual private networks (VPNs). You can find that information in the following documents:
  - *Any Transport over MPLS*

- *L2 VPN Interworking*

- The L2VPN Pseudowire Redundancy feature requires that the following mechanisms be in place to enable you to detect a failure in the network:
  - Label-switched paths (LSP) Ping/Traceroute and Any Transport over MPLS Virtual Circuit Connection Verification (AToM VCCV)
  - Local Management Interface (LMI)
  - Operation, Administration, and Maintenance (OAM)

Restrictions for L2VPN Pseudowire Redundancy

General Restrictions

- The primary and backup pseudowires must run the same type of transport service. The primary and backup pseudowires must be configured with AToM.
- Only static, on-box provisioning is supported.
- If you use L2VPN Pseudowire Redundancy with L2VPN Interworking, the interworking method must be the same for the primary and backup pseudowires.
- Setting the experimental (EXP) bit on the Multiprotocol Label Switching (MPLS) pseudowire is supported.
- Different pseudowire encapsulation types on the MPLS pseudowire are not supported.
- The **mpls l2transport route** command is not supported. Use the **xconnect** command instead.
- The ability to have the backup pseudowire fully operational at the same time that the primary pseudowire is operational is not supported. The backup pseudowire becomes active only after the primary pseudowire fails.
- The AToM VCCV feature is supported only on the active pseudowire.
- More than one backup pseudowire is not supported.
- Bidirectional Forwarding Detection over Virtual Circuit Connection Verification (BFDoVCCV) with status signaling is supported only on static pseudowires that do not have a backup peer. Explicit configuration of backup peers that violates this restriction is rejected.
- BFDoVCCV with status signaling through a pseudowire class is allowed. However, the feature is not supported on pseudowires that do not meet the restriction noted above.

Restrictions for Layer 2 Tunnel Protocol Version 3 (L2TPv3) Xconnect Configurations

- Interworking is not supported.
- Local switching backup by pseudowire redundancy is not supported.
- PPP, HDLC, and Frame-Relay attachment circuit (AC) types of L2TPv3 pseudowire redundancy are not supported.

- For the edge interface, only the Cisco 7600 series SPA Interface Processor-400 (SIP-400) linecard with the following shared port adapters (SPAs) is supported:

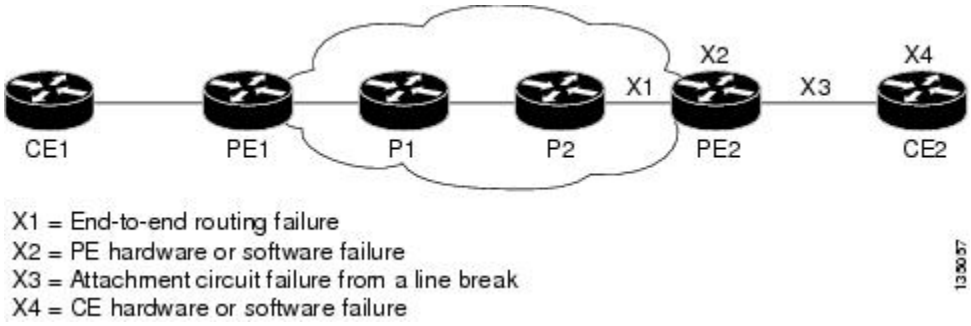
Cisco 2-Port Gigabit Ethernet Shared Port Adapter (SPA-2X1GE) Cisco 2-Port Gigabit Ethernet Shared Port Adapter, Version 2 (SPA-2X1GE-V2) Cisco 5-Port Gigabit Ethernet Shared Port Adapter, Version 2 (SPA-5X1GE-V2) Cisco 10-Port Gigabit Ethernet Shared Port Adapter, Version 2 (SPA-10X1GE-V2) Cisco 2-Port OC3c/STM1c ATM Shared Port Adapter (SPA-2XOC3-ATM) Cisco 4-Port OC3c/STM1c ATM Shared Port Adapter (SPA-4XOC3-ATM) Cisco 1-Port OC12c/STM4c ATM Shared Port Adapter (SPA-1XOC12-ATM) Cisco 1-Port OC-48c/STM-16 ATM Shared Port Adapter (SPA-1XOC48-ATM)

Information About L2VPN Pseudowire Redundancy

Introduction to L2VPN Pseudowire Redundancy

L2VPNs can provide pseudowire resiliency through their routing protocols. When connectivity between end-to-end PE devices fails, the L2VPN pseudowire redundancy can select and alternate path to the directed LDP session and the user data can take over. However, there are some parts of the network where this rerouting mechanism does not protect against interruptions in service. The figure below shows those parts of the network that are vulnerable to an interruption in service.

Figure 5: Points of Potential Failure in an L2VPN Network

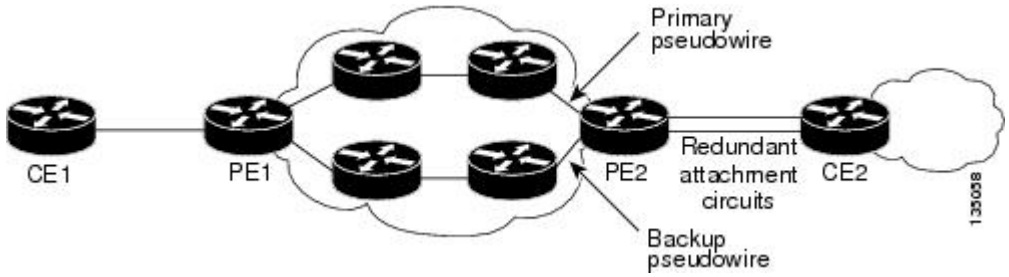


The L2VPN Pseudowire Redundancy feature provides the ability to ensure that the CE2 device in the figure above can always maintain network connectivity, even if one or all the failures in the figure occur.

The L2VPN Pseudowire Redundancy feature enables you to set up backup pseudowires. You can configure the network with redundant pseudowires (PWs) and redundant network elements, which are shown in the three figures below.

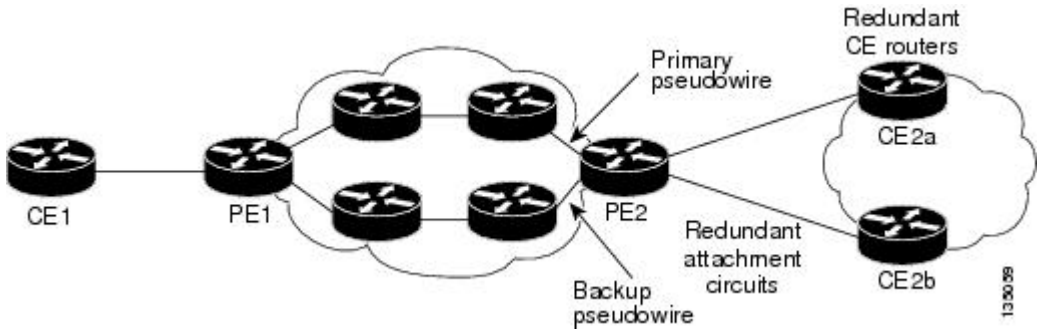
The figure below shows a network with redundant pseudowires and redundant attachment circuits.

Figure 6: L2VPN Network with Redundant PWs and Attachment Circuits



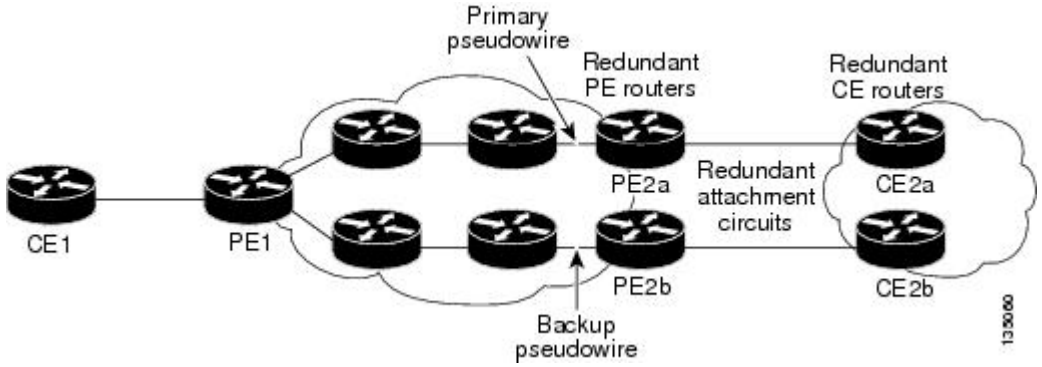
The figure below shows a network with redundant pseudowires, attachment circuits, and CE devices.

Figure 7: L2VPN Network with Redundant PWs, Attachment Circuits, and CE devices



The figure below shows a network with redundant pseudowires, attachment circuits, CE devices, and PE devices.

Figure 8: L2VPN Network with Redundant PWs, Attachment Circuits, CE devices, and PE devices



Xconnect as a Client of BFD

Redundant pseudowires are deployed to provide fault tolerance and resiliency to L2VPN-backhauled connections. The speed at which a system recovers from failures, especially when scaled to large numbers of

pseudowires, is critical to many service providers and service level agreements (SLAs). The configuration of a trigger for redundant pseudowire switchover reduces the time that it takes a large number of pseudowires to failover. A fundamental component of bidirectional forwarding detection (BFD) capability is enabled by fast-failure detection (FFD).

The configuration of this feature refers to a BFD configuration, such as the following (the second URL in the **bfd map** command is the loopback URL in the **monitor peer bfd** command):

```
bfd-template multi-hop mh
 interval min-tx 200 min-rx 200 multiplier 3 !
bfd map ipv4  10.1.1.0/24  10.1.1.1/32 mh
```

How to Configure L2VPN Pseudowire Redundancy

The L2VPN Pseudowire Redundancy feature enables you to configure a backup pseudowire in case the primary pseudowire fails. When the primary pseudowire fails, the PE router can switch to the backup pseudowire. You can have the primary pseudowire resume operation after it comes back up.

The default Label Distribution Protocol (LDP) session hold-down timer will enable the software to detect failures in about 180 seconds. That time can be configured so that the software can detect failures more quickly. See the **mpls ldp holdtime** command for more information.

Configuring the Pseudowire

The successful transmission of the Layer 2 frames between PE routers is due to the configuration of the PE routers. You set up the connection, called a pseudowire, between the routers.

The pseudowire-class configuration group specifies the characteristics of the tunneling mechanism, which are:

- Encapsulation type
- Control protocol
- Payload-specific options

You must specify the **encapsulation mpls**command as part of the pseudowire class for the AToM VCs to work properly. If you omit the **encapsulation mpls**command as part of the **xconnect**command, you receive the following error:

```
% Incomplete command.
```

Perform this task to configure a pseudowire class.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** name
4. **encapsulation mpls**
5. **interworking** {**ethernet** | **ip**}

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** name **Example:** `Router(config)# pseudowire-class atom`	Establishes a pseudowire class with a name that you specify. Enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation. For AToM, the encapsulation type is **mpls**.
Step 5	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw-class)# interworking ip`	(Optional) Enables the translation between the different Layer 2 encapsulations.

Configuring L2VPN Pseudowire Redundancy

Use the following steps to configure the L2VPN Pseudowire Redundancy feature.

Before You Begin

For each transport type, the **xconnect**command is configured slightly differently. The following configuration steps use Ethernet VLAN over MPLS, which is configured in subinterface configuration mode. See *Any Transport over MPLS* to determine how to configure the **xconnect** command for other transport types.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *subslot* / *interface* **.** *subinterface*
4. **encapsulation dot1q** vlan-id
5. **xconnect** *peer-router-id vcid* {**encapsulation mpls**| **pw-class** *pw-class-name}*
6. **backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*]
7. **backup delay** *e nable-delay* {*disable-delay* | **never**}

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 gigabitethernet** *slot* / *subslot* / *interface* **.** *subinterface* **Example:** `Router(config)# interface gigabitethernet0/0/0.1`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. Make sure that the subinterface on the adjoining CE router is on the same VLAN as this PE router.
Step 4	**encapsulation dot1q** vlan-id **Example:** `Router(config-subif)# encapsulation dot1q 100`	Enables the subinterface to accept 802.1Q VLAN packets. The subinterfaces between the CE and PE routers that are running Ethernet over MPLS must be in the same subnet. All other subinterfaces and backbone routers do not.
Step 5	**xconnect** *peer-router-id vcid* {**encapsulation mpls**| **pw-class** *pw-class-name}* **Example:** `Router(config-subif)# xconnect 10.0.0.1 123 pw-class atom`	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports. Enters xconnect configuration mode.
Step 6	**backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*]	Specifies a redundant peer for the pseudowire VC.

	Command or Action	Purpose
	Example: `Router(config-if-xconn)# backup peer 10.0.0.3 125 pw-class atom`	The pseudowire class name must match the name you specified when you created the pseudowire class, but you can use a different pw-class in the **backup peer** command than the name that you used in the primary **xconnect** command.
Step 7	**backup delay** *e nable-delay* {*disable-delay* | **never**} **Example:** `Router(config-if-xconn)# backup delay 5 never`	Specifies how long (in seconds) the backup pseudowire VC should wait to take over after the primary pseudowire VC goes down. The range is 0 to 180. Specifies how long the primary pseudowire should wait after it becomes active to take over for the backup pseudowire VC. The range is 0 to 180 seconds. If you specify the **never keyword**, the primary pseudowire VC never takes over for the backup.

Configuring Xconnect as a Client of BFD

Perform this task to configure a trigger for redundant pseudowire switchover.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class mpls-ffd**
   - Enters pseudowire class configuration mode.
4. **encapsulation mpls**
5. **monitor peer bfd** [**local interface** *interface-type interface-number*]

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	**pseudowire-class mpls-ffd** • Enters pseudowire class configuration mode. **Example:** `Device(config)# pseudowire-class mpls-ffd`	Establishes a pseudowire class for MPLS fast-failure detection.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation to be MPLS.
Step 5	**monitor peer bfd** [**local interface** *interface-type interface-number*] **Example:** `Device(config-pw-class)# monitor peer bfd local interface loopback 0`	Enables the pseudowire fast-failure detection capability.

Forcing a Manual Switchover to the Backup Pseudowire VC

To force the router switch over to the backup or primary pseudowire, you can enter the **xconnect backup force switchover** command in privileged EXEC mode. You can specify either the interface of the primary attachment circuit (AC) to switch to or the IP-address and VC ID of the peer router.

A manual switchover can be made only if the interface or peer specified in the command is actually available and the xconnect will move to the fully active state when the command is entered.

SUMMARY STEPS

1. **enable**
2. **xconnect backup force-switchover {** **interface** *interface-info* | **peer** *ip-address vcid}*

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	**xconnect backup force-switchover { interface** *interface-info* | **peer** *ip-address vcid}* **Example:** `Router# xconnect backup force-switchover peer 10.10.10.1 123`	Specifies that the router should switch to the backup or to the primary pseudowire.

Verifying the L2VPN Pseudowire Redundancy Configuration

Use the following commands to verify that the L2VPN Pseudowire Redundancy feature is correctly configured.

SUMMARY STEPS

1. **show mpls l2transport vc**
2. **show xconnect all**
3. **xconnect logging redundancy**

DETAILED STEPS

Step 1 **show mpls l2transport vc**

In this example, the primary attachment circuit is up. The backup attachment circuit is available, but not currently selected. The **show** output displays as follows:

Example:

```
Router# show mpls l2transport vc
Local intf     Local circuit           Dest address    VC ID      Status
-------------  ----------------------- --------------- ---------- ----------
Et0/0.1        Eth VLAN 101            10.0.0.2        101        UP
Et0/0.1        Eth VLAN 101            10.0.0.3        201        DOWN

Router# show mpls l2transport vc detail
Local interface: Et0/0.1 up, line protocol up, Eth VLAN 101 up
   Destination address 10.0.0.2 VC ID: 101, VC status UP
   .
   .
   .
Local interface: Et0/0.1 down, line protocol down, Eth VLAN 101 down
   Destination address 10.0.0.3 VC ID: 201, VC status down
   .
   .
   .
```

Step 2 **show xconnect all**

In this example, the topology is Attachment Circuit 1 to Pseudowire 1 with a Pseudowire 2 as a backup:

Example:

```
Router# show xconnect all
Legend: XC ST=Xconnect State, S1=Segment1 State, S2=Segment2 State
UP=Up, DN=Down, AD=Admin Down, IA=Inactive, NH=No Hardware
XC ST  Segment 1                         S1 Segment 2                         S2
------+------------------------------------+--+------------------------------------+--
UP pri ac   Et0/0(Ethernet)              UP mpls 10.55.55.2:1000              UP
IA sec ac   Et0/0(Ethernet)              UP mpls 10.55.55.3:1001              DN
```

In this example, the topology is Attachment Circuit 1 to Attachment Circuit 2 with a Pseudowire backup for Attachment Circuit 2:

Example:

```
Router# show xconnect all
Legend: XC ST=Xconnect State, S1=Segment1 State, S2=Segment2 State
UP=Up, DN=Down, AD=Admin Down, IA=Inactive, NH=No Hardware
XC ST  Segment 1                         S1 Segment 2                         S2
------+------------------------------------+--+------------------------------------+--
UP pri ac   Se6/0:150(FR DLCI)           UP ac   Se8/0:150(FR DLCI)           UP
IA sec ac   Se6/0:150(FR DLCI)           UP mpls 10.55.55.3:7151              DN
```

Step 3 **xconnect logging redundancy**

In addition to the **show mpls l2transport vc**command and the **show xconnect** command, you can use the **xconnect logging redundancy** command to track the status of the xconnect redundancy group:

Example:

```
Router(config)# xconnect logging redundancy
```

When this command is configured, the following messages will be generated during switchover events:

Activating the primary member:

Example:

```
00:01:07: %XCONNECT-5-REDUNDANCY: Activating primary member 10.55.55.2:1000
```

Activating the backup member:

Example:

```
00:01:05: %XCONNECT-5-REDUNDANCY: Activating secondary member 10.55.55.3:1001
```

Configuration Examples for L2VPN Pseudowire Redundancy

Each of the configuration examples refers to one of the following pseudowire classes:

- AToM (like-to-like) pseudowire class:

```
pseudowire-class mpls
 encapsulation mpls
```

- L2VPN IP interworking:

```
pseudowire-class mpls-ip
 encapsulation mpls
 interworking ip
```

L2VPN Pseudowire Redundancy and AToM Like to Like Examples

The following example shows a High-Level Data Link Control (HDLC) attachment circuit xconnect with a backup pseudowire:

```
interface Serial4/0
 xconnect 10.55.55.2 4000 pw-class mpls
 backup peer 10.55.55.3 4001 pw-class mpls
```

The following example shows a Frame Relay attachment circuit xconnect with a backup pseudowire:

```
connect fr-fr-pw Serial6/0 225 l2transport
 xconnect 10.55.55.2 5225 pw-class mpls
 backup peer 10.55.55.3 5226 pw-class mpls
```

L2VPN Pseudowire Redundancy and L2VPN Interworking Examples

The following example shows an Ethernet attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
interface Ethernet0/0
 xconnect 10.55.55.2 1000 pw-class mpls-ip
 backup peer 10.55.55.3 1001 pw-class mpls-ip
```

The following example shows an Ethernet VLAN attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
interface Ethernet1/0.1
 encapsulation dot1Q 200
 no ip directed-broadcast
 xconnect 10.55.55.2 5200 pw-class mpls-ip
 backup peer 10.55.55.3 5201 pw-class mpls-ip
```

The following example shows a Frame Relay attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
connect fr-ppp-pw Serial6/0 250 l2transport
 xconnect 10.55.55.2 8250 pw-class mpls-ip
 backup peer 10.55.55.3 8251 pw-class mpls-ip
```

The following example shows a PPP attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
interface Serial7/0
 encapsulation ppp
 xconnect 10.55.55.2 2175 pw-class mpls-ip
 backup peer 10.55.55.3 2176 pw-class mpls-ip
```

L2VPN Pseudowire Redundancy with Layer 2 Local Switching Examples

The following example shows an Ethernet VLAN-VLAN local switching xconnect with a pseudowire backup for Ethernet segment E2/0.2. If the subinterface associated with E2/0.2 goes down, the backup pseudowire is activated.

```
connect vlan-vlan Ethernet1/0.2 Ethernet2/0.2
 backup peer 10.55.55.3 1101 pw-class mpls
```

The following example shows a Frame Relay-to-Frame Relay local switching connect with a pseudowire backup for Frame Relay segment S8/0 150. If data-link connection identifier (DLCI) 150 on S8/0 goes down, the backup pseudowire is activated.

```
connect fr-fr-ls Serial6/0 150 Serial8/0 150
 backup peer 10.55.55.3 7151 pw-class mpls
```

Additional References

Related Documents

Related Topic	Document Title
Any Transport over MPLS	Any Transport over MPLS
High Availability for AToM	AToM Graceful Restart
L2VPN Interworking	L2VPN Interworking
Layer 2 local switching	Layer 2 Local Switching
PWE3 MIB	Pseudowire Emulation Edge-to-Edge MIBs for Ethernet and Frame Relay Services
Packet sequencing	Any Transport over MPLS (AToM) Sequencing Support
BFD configuration	IP Routing BFD Configuration Guide

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

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 L2VPN Pseudowire Redundancy

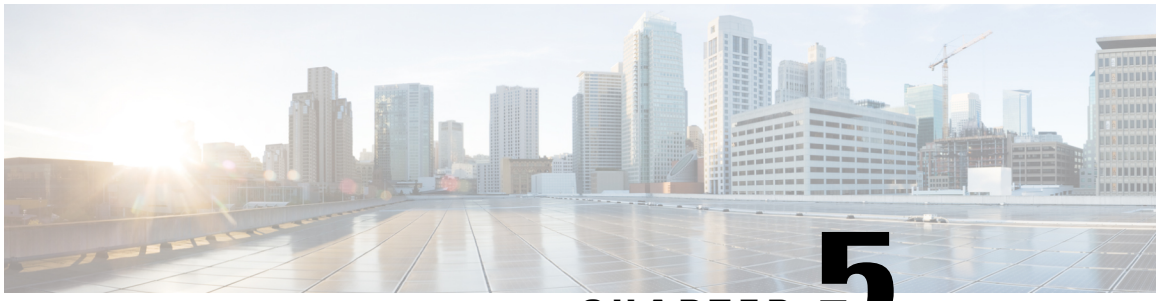
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train 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 17: Feature Information for L2VPN Pseudowire Redundancy

Feature Name	Releases	Feature Information
L2VPN Pseudowire Redundancy	12.0(31)S 12.2(28)SB 12.2(22)SXI 12.2(33)SRB 12.4(11)T 15.0(1)S	This feature enables you to set up your network to detect a failure in the network and reroute the Layer 2 service to another endpoint that can continue to provide service. In Cisco IOS Release 12.0(31)S, the L2VPN Pseudowire Redundancy feature was introduced for Any Transport over MPLS (AToM) on the Cisco 12000 series routers. This feature was integrated into Cisco IOS Release 12.2(28)SB. This feature was integrated into Cisco IOS Release 12.4(11)T. This feature was integrated into Cisco IOS Release 12.2(33)SRB. This feature was integrated into Cisco IOS Release 12.2(33)SXI. The following commands were introduced or modified: **backup delay (L2VPN local switching)**, **backup peer**, **show xconnect**, **xconnect backup force-switchover**, **xconnect logging redundancy**.
L2VPN Pseudowire Redundancy for L2TPv3	12.2(33)SRE 15.0(1)S	This feature provides L2VPN pseudowire redundancy for L2TPv3 xconnect configurations. In Cisco IOS Release 12.2(33)SRE, this feature was implemented on the Cisco 7600 series routers.
Xconnect as a Client of BFD	15.1(3)S	This feature provides fast-failure detection for L2VPN pseudowire redundancy. The following command was introduced: **monitor peer bfd**.

Feature Name	Releases	Feature Information
Resilient Pseudowire (RPW): PW Fast Recovery	15.2(1)S	This feature was integrated into Cisco IOS Release 15.2(1)S. The following commands were introduced or modified: **aps hspw-icrm-grp** , **show hspw-aps-icrm**.



CHAPTER 5

L2VPN Pseudowire Switching

This feature module explains how to configure L2VPN Pseudowire Switching, which extends Layer 2 Virtual Private Network (L2VPN) pseudowires across an interautonomous system (inter-AS) boundary or across two separate Multiprotocol Label Switching (MPLS) networks. The feature supports ATM and time-division multiplexing (TDM) attachment circuits (ACs) and Ethernet ACs.

Finding Feature Information

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

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

Prerequisites for L2VPN Pseudowire Switching

For the Cisco 12000 series routers, the L2VPN Pseudowire Switching feature for Any Transport over MPLS (AToM) is supported on the following engines:

- E2

- E3
- E4+
- E5
- E6

For engines that do not support this feature, the packets are sent to the software and forwarded through the slow path.

Note Engines E1 and E4 do not support L2VPN Pseudowire Switching, even in the slow path.

Restrictions for L2VPN Pseudowire Switching

- L2VPN Pseudowire Switching is supported with AToM.
- Only static, on-box provisioning is supported.
- Sequencing numbers in AToM packets are not processed by L2VPN Pseudowire Switching. The feature blindly passes the sequencing data through the xconnect packet paths, a process that is called transparent sequencing. The endpoint provider-edge (PE) to customer-edge (CE) connections enforce the sequencing.
- You can ping the adjacent next-hop PE router. End-to-end label switched path (LSP) pings are not supported.
- Do not configure IP or Ethernet interworking on a router where L2VPN Pseudowire Switching is enabled. Instead, configure interworking on the routers at the edge PEs of the network.
- The control word negotiation results must match. If either segment does not negotiate the control word, the control word is disabled for both segments.
- AToM Graceful Restart is negotiated independently on each pseudowire segment. If there is a transient loss of the label distribution protocol (LDP) session between two AToM PE routers, packets continue to flow.
- Per-pseudowire quality of service (QoS) is not supported. Traffic engineering (TE) tunnel selection is supported.
- Attachment circuit interworking is not supported.

Information About L2VPN Pseudowire Switching

How L2VPN Pseudowire Switching Works

L2VPN Pseudowire Switching allows the user to extend L2VPN pseudowires across two separate MPLS networks or across an inter-AS boundary, as shown in the two figures below.

L2VPN Pseudowire Switching connects two or more contiguous pseudowire segments to form an end-to-end multihop pseudowire. This end-to-end pseudowire functions as a single point-to-point pseudowire.

As shown in the second figure below, L2VPN Pseudowire Switching enables you to keep the IP addresses of the edge PE routers private across inter-AS boundaries. You can use the IP address of the Autonomous System Boundary Routers (ASBRs) and treat them as pseudowire aggregation (PE-agg) routers. The ASBRs join the pseudowires of the two domains.

L2VPN Pseudowire Switching also enables you to keep different administrative or provisioning domains to manage the end-to-end service. At the boundaries of these networks, PE-agg routers delineate the management responsibilities.

Figure 9: L2VPN Pseudowire Switching in an Intra-AS Topology

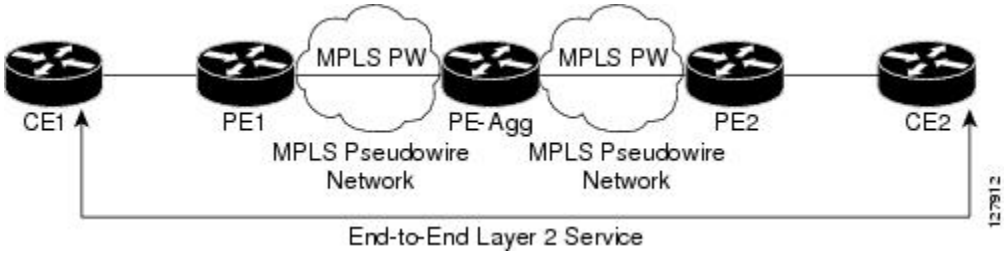
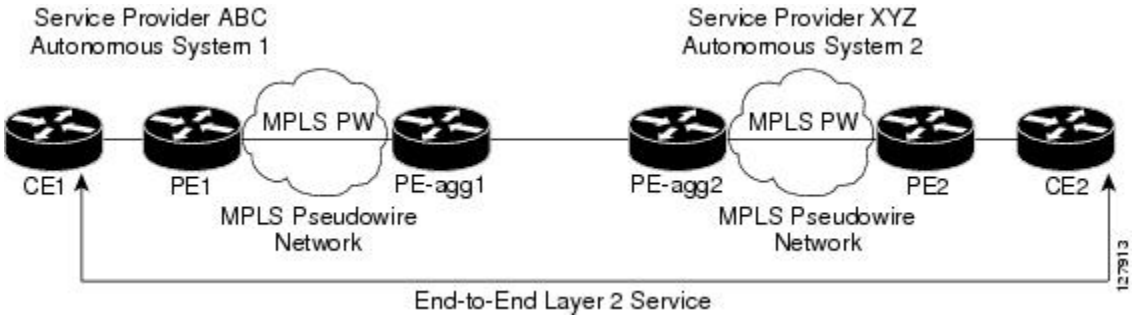


Figure 10: L2VPN Pseudowire Switching in an Inter-AS Topology



How Packets Are Manipulated at the L2VPN Pseudowire Switching Aggregation Point

Switching AToM packets between two AToM pseudowires is the same as switching any MPLS packet. The MPLS switching data path switches AToM packets between two AToM pseudowires. The following list explains exceptions:

- The outgoing virtual circuit (VC) label replaces the incoming VC label in the packet. New Internal Gateway Protocol (IGP) labels and Layer 2 encapsulation are added.
- The incoming VC label time-to-live (TTL) field is decremented by one and copied to the outgoing VC label TTL field.
- The incoming VC label EXP value is copied to the outgoing VC label EXP field.
- The outgoing VC label “Bottom of Stack” S bit in the outgoing VC label is set to 1.

- AToM control word processing is not performed at the L2VPN Pseudowire Switching aggregation point. Sequence numbers are not validated. Use the Router Alert label for LSP Ping; do not require control word inspection to determine an LSP Ping packet.

How to Configure L2VPN Pseudowire Switching

Use the following procedure to configure L2VPN Pseudowire Switching on each of the PE-agg routers. In this configuration, you are limited to two **neighbor**commands after entering the **l2 vfi**command.

Before You Begin

- This procedure assumes that you have configured basic AToM L2VPNs. This procedure does not explain how to configure basic AToM L2VPNs that transport Layer 2 packets over an MPLS backbone. For information on the basic configuration, see Any Transport over MPLS .
- For interautonomous configurations, ASBRs require a labeled interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *name* **point-to-point**
4. **neighbor** *ip-address vcid* [**encapsulation mpls** | **pw-class** *pw-class-name]*
5. **exit**
6. **exit**
7. **show mpls l2transport vc** [**vcid** [*vc-id* | *vc-id-min vc-id-max*]] [**interface** *name*[*local-circuit-id*]] [**destination** *ip-address* | *name*] [**detail**]
8. **show vfi** [*vfi-name*]
9. **ping** [*protocol*] [**tag**] {*host-name*| *system-address*}

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	**l2 vfi** *name* **point-to-point** **Example:** `Router(config)# l2 vfi atomtunnel point-to-point`	Creates a point-to-point Layer 2 virtual forwarding interface (VFI) and enters VFI configuration mode.
Step 4	**neighbor** *ip-address vcid* [**encapsulation mpls** | **pw-class** *pw-class-name]* **Example:** `Router(config-vfi)# neighbor 10.0.0.1 100 pw-class mpls`	Configures 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	**exit** **Example:** `Router(config-vfi)# exit`	Exits VFI configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 7	**show mpls l2transport vc** [**vcid** [*vc-id* | *vc-id-min vc-id-max*]] [**interface** *name*[*local-circuit-id*]] [**destination** *ip-address* | *name*] [**detail**] **Example:** `Router# show mpls l2transport vc`	Verifies that the L2VPN Pseudowire Switching session has been established.
Step 8	**show vfi** [*vfi-name*] **Example:** `Router# show vfi atomtunnel`	Verifies that a point-to-point VFI has been established.
Step 9	**ping** [*protocol*] [**tag**] {*host-name*| *system-address*} **Example:** `Router# ping 10.1.1.1`	When issued from the CE routers, verifies end-to-end connectivity.

Examples

The following example displays output from the **show mpls l2transport vc** command:

```
Router# show mpls l2transport vc
Local intf     Local circuit              Dest address    VC ID Status
-------------  -------------------------- --------------- ----- ----
MPLS PW        10.0.1.1:100               10.0.1.1        100   UP
MPLS PW        10.0.1.1:100               10.0.1.1        100   UP
```

The following example displays output from the **show vfi**command:

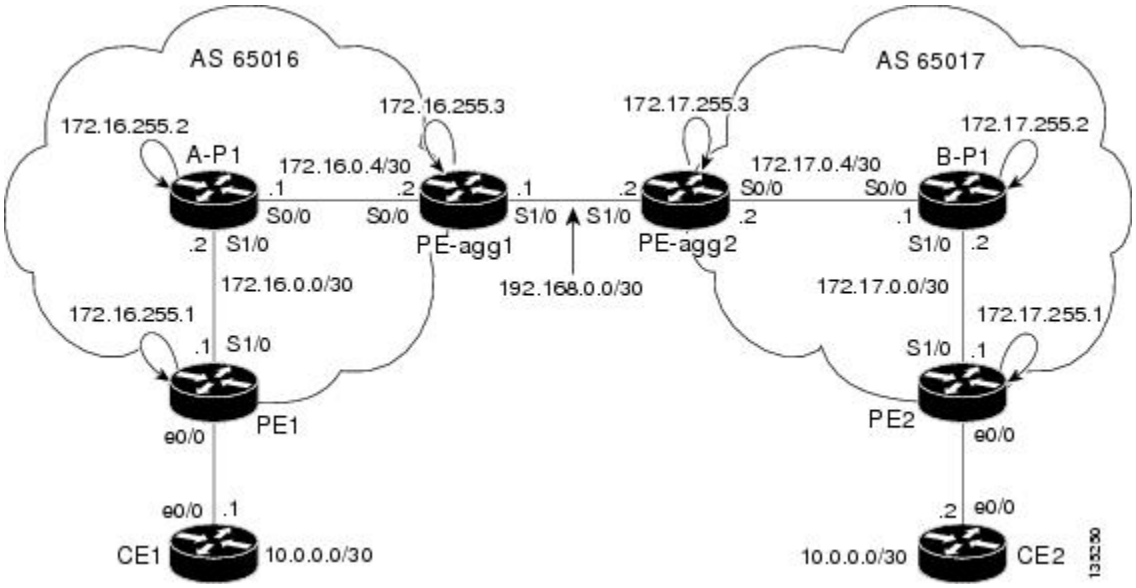
```
Router# show vfi
VFI name: test, type: point-to-point
 Neighbors connected via pseudowires:
   Router ID        Pseudowire ID
   10.0.1.1         100
   10.0.1.1         100
```

Configuration Examples for L2VPN Pseudowire Switching

L2VPN Pseudowire Switching in an Inter-AS Configuration Example

Two separate autonomous systems are able to pass L2VPN packets, because the two PE-agg routers have been configured with L2VPN Pseudowire Switching. This example configuration is shown in the figure below.

Figure 11: L2VPN Pseudowire Switching in an Interautonomous System



PE-agg-1	PE-agg-2

PE-agg-1	PE-agg-2
version 12.0	version 12.0
service timestamps debug uptime	service timestamps debug uptime
service timestamps log uptime	service timestamps log uptime
service password-encryption	service password-encryption
!	!
hostname [pe-agg1]	hostname [pe-agg2]
!	!
boot-start-marker	boot-start-marker
boot-end-marker	boot-end-marker
!	!
enable secret 5 1Q0Bb$32sIU82pHRgyddWaeB4zs/	enable secret 5 $1$32jd$zQRfxXzjstr4llV9DcWf7/
!	!
ip subnet-zero	ip subnet-zero
ip cef	ip cef
no ip domain-lookup	no ip domain-lookup
mpls label protocol ldp	mpls label protocol ldp
pseudowire-class SW-PW	pseudowire-class SW-PW
encapsulation mpls	encapsulation mpls
!	!
l2 vfi PW-SWITCH-1 point-to-point	l2 vfi PW-SWITCH-1 point-to-point
neighbor 172.17.255.3 100 pw-class SW-PW	neighbor 172.16.255.3 100 pw-class SW-PW
neighbor 172.16.255.1 16 pw-class SW-PW	neighbor 172.17.255.1 17 pw-class SW-PW
!	!
interface Loopback0	interface Loopback0
ip address 172.16.255.3 255.255.255.255	ip address 172.17.255.3 255.255.255.255
no ip directed-broadcast	no ip directed-broadcast
!	!
interface Serial0/0	interface Serial0/0
ip address 172.16.0.6 255.255.255.252	ip address 172.17.0.6 255.255.255.252
no ip directed-broadcast	no ip directed-broadcast
mpls ip	mpls ip
!	!
interface Serial1/0	interface Serial1/0
ip address 192.168.0.1 255.255.255.252	ip address 192.168.0.2 255.255.255.252

PE-agg-1	PE-agg-2
no ip directed-broadcast	no ip directed-broadcast
mpls bgp forwarding	mpls bgp forwarding
!	!
router ospf 16	router ospf 17
log-adjacency-changes	log-adjacency-changes
network 172.16.0.0 0.0.255.255 area 0	network 172.17.0.0 0.0.255.255 area 0
!	!
router bgp 65016	router bgp 65017
no synchronization	no synchronization
bgp log-neighbor-changes	bgp log-neighbor-changes
network 172.16.255.3 mask 255.255.255.255	network 172.17.255.3 mask 255.255.255.255
neighbor 192.168.0.2 remote-as 65017	neighbor 192.168.0.1 remote-as 65016
neighbor 192.168.0.2 send-label	neighbor 192.168.0.1 send-label
no auto-summary	no auto-summary
!	!
ip classless	ip classless
control-plane	control-plane
!	!
line con 0	line con 0
exec-timeout 0 0	exec-timeout 0 0
line aux 0	line aux 0
line vty 0 4	line vty 0 4
login	login
!	!
no cns aaa enable	no cns aaa enable
end	end

A-P1	B-P1

A-P1	B-P1
version 12.0 service timestamps debug uptime service timestamps log uptime service password-encryption ! hostname [a-p1] ! boot-start-marker boot-end-marker ! enable secret 5 1eiUn$rTMnZiYnJxtMTpO0NKpQQ/ ! ip subnet-zero ip cef no ip domain-lookup mpls label protocol ldp ! interface Loopback0 ip address 172.16.255.2 255.255.255.255 no ip directed-broadcast ! interface Serial0/0 ip address 172.16.0.5 255.255.255.252 no ip directed-broadcast mpls ip ! interface Serial1/0 ip address 172.16.0.2 255.255.255.252 no ip directed-broadcast mpls ip ! router ospf 16 log-adjacency-changes network 172.16.0.0 0.0.255.255 area 0	version 12.0 service timestamps debug uptime service timestamps log uptime service password-encryption ! hostname [b-p1] ! boot-start-marker boot-end-marker ! enable secret 5 1svU/$2JmJZ/5gxlW4nVXVniIJe1 ! ip subnet-zero ip cef no ip domain-lookup mpls label protocol ldp ! interface Loopback0 ip address 172.17.255.2 255.255.255.255 no ip directed-broadcast ! interface Serial0/0 ip address 172.17.0.5 255.255.255.252 no ip directed-broadcast mpls ip ! interface Serial1/0 ip address 172.17.0.2 255.255.255.252 no ip directed-broadcast mpls ip ! router ospf 17 log-adjacency-changes network 172.17.0.0 0.0.255.255 area 0

A-P1	B-P1
!	!
ip classless	ip classless
!	!
control-plane	control-plane
!	!
line con 0	line con 0
exec-timeout 0 0	exec-timeout 0 0
line aux 0	line aux 0
line vty 0 4	line vty 0 4
login	login
!	!
no cns aaa enable	no cns aaa enable
end	end

PE1	PE2

PE1	PE2
version 12.0	version 12.0
service timestamps debug uptime	service timestamps debug uptime
service timestamps log uptime	service timestamps log uptime
service password-encryption	service password-encryption
!	!
hostname [pe1]	hostname [pe2]
!	!
boot-start-marker	boot-start-marker
boot-end-marker	boot-end-marker
!	!
enable secret 5 $1$9z8F$2A1/YLc6NB6d.WLQXF0Bz1	enable secret 5 1rT.V$8Z6Dy/r8/eaRdx2TR/O5r/
!	!
ip subnet-zero	ip subnet-zero
ip cef	ip cef
no ip domain-lookup	no ip domain-lookup
mpls label protocol ldp	mpls label protocol ldp
pseudowire-class ETH-PW	pseudowire-class ETH-PW
encapsulation mpls	encapsulation mpls
!	!
interface Loopback0	interface Loopback0
ip address 172.16.255.1 255.255.255.255	ip address 172.17.255.1 255.255.255.255
no ip directed-broadcast	no ip directed-broadcast
!	!
interface Ethernet0/0	interface Ethernet0/0
no ip address	no ip address
no ip directed-broadcast	no ip directed-broadcast
no cdp enable	no cdp enable
xconnect 172.16.255.3 16 pw-class ETH-PW	xconnect 172.17.255.3 17 pw-class ETH-PW
!	!
interface Serial1/0	interface Serial1/0
ip address 172.16.0.1 255.255.255.252	ip address 172.17.0.1 255.255.255.252
no ip directed-broadcast	no ip directed-broadcast
mpls ip	mpls ip
!	!

PE1	PE2
router ospf 16 log-adjacency-changes network 172.16.0.0 0.0.255.255 area 0 ! ip classless ! control-plane ! line con 0 exec-timeout 0 0 line aux 0 line vty 0 4 login ! no cns aaa enable end	router ospf 17 log-adjacency-changes network 172.17.0.0 0.0.255.255 area 0 ! ip classless ! control-plane ! line con 0 exec-timeout 0 0 line aux 0 line vty 0 4 login ! no cns aaa enable end

CE1	CE2

CE1	CE2
version 12.0 service timestamps debug uptime service timestamps log uptime service password-encryption ! hostname [ce1] ! boot-start-marker boot-end-marker ! enable secret 5 1o9N6$LSrxHufTn0vjCY0nW8hQX. ! ip subnet-zero ip cef no ip domain-lookup ! interface Ethernet0/0 ip address 10.0.0.1 255.255.255.252 no ip directed-broadcast ! ip classless ! control-plane ! line con 0 exec-timeout 0 0 line aux 0 line vty 0 4 login ! no cns aaa enable end	version 12.0 service timestamps debug uptime service timestamps log uptime service password-encryption ! hostname [ce2] ! boot-start-marker boot-end-marker ! enable secret 5 1YHo6$LQ4z5PdrF5B9dnL75Xvvm1 ! ip subnet-zero ip cef no ip domain-lookup ! interface Ethernet0/0 ip address 10.0.0.2 255.255.255.252 no ip directed-broadcast ! ip classless ! control-plane ! line con 0 exec-timeout 0 0 line aux 0 line vty 0 4 login ! no cns aaa enable end

Additional References

Related Documents

Related Topic	Document Title
Any Transport over MPLS	Any Transport over MPLS
Pseudowire redundancy	http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s31/fsstitch.htm *L2VPN Pseudowire Redundancy*
High availability for AToM	AToM Graceful Restart
L2VPN interworking	L2VPN Interworking
Layer 2 local switching	Layer 2 Local Switching
PWE3 MIB	Pseudowire Emulation Edge-to-Edge MIBs for Ethernet and Frame Relay Services
Packet sequencing	Any Transport over MPLS (AToM) Sequencing Support

Standards

Standard	Title
draft-ietf-pwe3-control-protocol-14.txt	*Pseudowire Setup and Maintenance using LDP*
draft-martini-pwe3-pw-switching-01.txt	*Pseudo Wire Switching*

MIBs

MIB	MIBs Link
• CISCO-IETF-PW-MIB • CISCO-IETF-PW-MPLS-MIB • CISCO-IETF-PW-ENET-MIB • CISCO-IETF-PW-FR-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

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/cisco/web/support/index.html

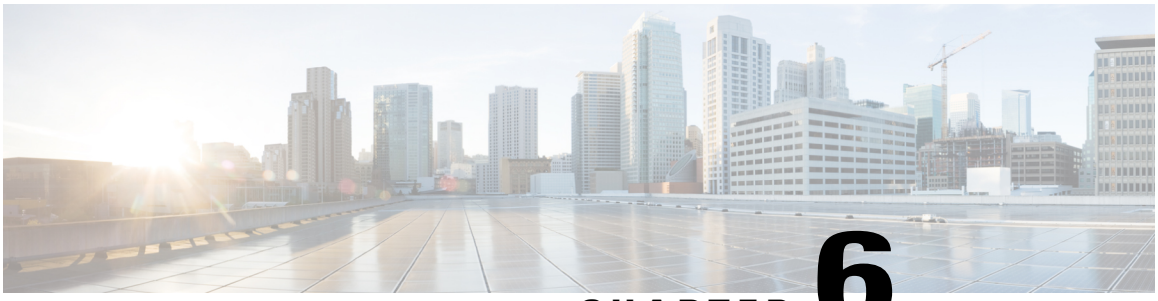
Feature Information for L2VPN Pseudowire Switching

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 18: Feature Information for L2VPN Pseudowire Switching

Feature Name	Releases	Feature Information
L2VPN Pseudowire Switching	12.0(31)S, 12.2(28)SB, 12.2(33)SRB, 12.2(33)SRD2, 12.2(33)SRE	This feature configures L2VPN Pseudowire Switching, which extends L2VPN pseudowires across an interautonomous system (inter-AS) boundary or across two separate MPLS networks. In Cisco IOS Release 12.2(28)SB, support was added for the Cisco 7200 and 7301 series routers. In 12.2(33)SRD2, support was added for ATM and TDM ACs. The following commands were introduced or modified: **l2 vfi point-to-point**, **neighbor**(L2VPN Pseudowire Switching), **show vfi**.



CHAPTER 6

L2VPN Advanced VPLS

The L2VPN Advanced VPLS feature introduces the following enhancements to Virtual Private LAN Services:

- Ability to load-balance traffic across multiple core interfaces using equal cost multipaths (ECMP)
- Support for redundant provide edge switches
- Command line interface enhancements to facilitate configuration of the L2VPN Advanced VPLS feature

The L2VPN Advanced VPLS feature uses Virtual Switch System (VSS) and Flow Aware Transport (FAT) pseudowires to achieve PE redundancy and load-balancing. The following sections explain the concepts and configuration tasks for this feature.

Finding Feature Information

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

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

Prerequisites for L2VPN Advanced VPLS

- This feature requires that you understand how VPLS works. For information about VPLS, see "VPLS Overview" section in the *Cisco 7600 Series Ethernet Services Plus (ES+) and Ethernet Services Plus T (ES+T) Line Card* Configuration Guide.
- Configuring the L2VPN Advanced VPLS feature works with MPLS Traffic Engineering tunnels with explicit paths and Generic Routing Encapsulation (GRE tunnels) with static routes to the tunnel destination. For information and configuration steps for MPLS traffic engineering and GRE tunnels, see the following documents:
  - MPLS Traffic Engineering and Enhancements
  - Implementing Tunnels
- This features requires two Cisco 6500 series routers be configured as a virtual switch system.
- This features requires nonstop forwarding and stateful switchover.

Restrictions for L2VPN Advanced VPLS

- The **ping** and **traceroute** commands that support the Any Transport over MPLS Virtual Circuit Connection Verification (VCCV) feature are not supported over FAT pseudowires.
- The VPLS Autodiscovery feature is not supported with the L2VPN Advanced VPLS feature.
- In Cisco IOS Release 12.2(33)SXI4, the following types of configurations are supported:
  - MPLS core with configuration of PE routers through the **neighbor** command under transport vpls mode.
  - MPLS core with configuration of PE routers through MPLS traffic engineering tunnels using explicit paths.
  - IP core with configuration of PE routers through MPLS over GRE tunnels.

Other configuration methods, including using the **route-via** command, BGP autodiscovery, or explicit VLAN assignment to a PE egress port, are not supported.

- Load-balancing is not supported in the core routers when the core uses IP to transport packets.
- The maximum number of links per bundle is limited to eight.
- The maximum number of port channels is limited to 32.
- The maximum number of VPLS neighbors is limited to 60 minus the number of neighbors configured with the **load-balanceflow** command.
- In Cisco IOS Release 12.2(33)SXI4, the L2VPN Advanced VPLS feature is supported on the Cisco Catalyst 6500 series switches with Supervisor 720-10GE engine.
- The L2VPN Advanced VPLS feature supports the following line cards and shared port adapters (SPAs):
  - 7600-SIP-400 (core facing)

- Gigabit and 10-gigabit Ethernet SPAs (2X1GE-V1, 2X1GE-V2 and 1X10GE-V2 SPA)
- Packet over Sonet (POS) SPAs (2XOC3, 4XOC3, 1XOC12 and 1XOC48)

Information About L2VPN Advanced VPLS

FAT Pseudowires and Their Role in Load-Balancing

FAT pseudowires are used to load-balance traffic in the core when equal cost multipaths are used. The MPLS labels add an additional label to the stack, called the flow label, which contains the flow information of a VC. For more information about FAT pseudowires, see PWE3 Internet-Draft *Flow Aware Transport of MPLS Pseudowires* (draft-bryant-filsfils-fat-pw).

Virtual Switch Systems

Two Cisco 6500 series switches can be connected to form one logical switch. One switch is designated as the master, while the other is the slave. The two switches are connected by a virtual switch link (VSL). The two switches are used for link redundancy, load-balancing, and failover.

For more information on virtual switch systems, see the "Configuring VSS" section in the *Catalyst 6500 Release 12.2SXH and Later Software Configuration Guide*.

How to Configure L2VPN Advanced VPLS

Enabling Load-Balancing with ECMP and FAT Pseudowires

The following steps explain how to enable load-balancing at the provider edge (PE) device and on the core device.

To enable load-balancing on the edge device, issue the **load-balance flow** command. The load-balancing rules are configured through the **port-channel load-balance** command parameters.

To enable core load-balancing, issue the **flow-label enable** command on both PE devices. You must issue the **load-balance flow** command with the **flow-label enable** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *name*
4. **encapsulation mpls**
5. **load-balance flow**
6. **flow-label enable**
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	**pseudowire-class** *name* **Example:** `Device(config)# pseudowire-class class1`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw)# encapsulation mpls`	Specifies the MPLS tunneling encapsulation type.
Step 5	**load-balance flow** **Example:** `Device(config-pw)# load-balance flow`	Enables load-balancing on ECMPs.
Step 6	**flow-label enable** **Example:** `Device(config-pw)# flow-label enable`	Enables the imposition and disposition of flow labels for the pseudowire.
Step 7	**end** **Example:** `Device(config-pw)# end`	Exits pseudowire class configuration mode and enters privileged EXEC mode.

Enabling Port-Channel Load-Balancing

The following task explains how to enable port channel load-balancing, which sets the load-distribution method among the ports in the bundle. If the **port-channel load-balance** command is not configured, load-balancing occurs with default parameters.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **port-channel load-balance** *method*
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	**port-channel load-balance** *method* **Example:** `Device(config)# port-channel load-balance src-mac`	Specifies the load distribution method among the ports in a bundle.
Step 4	exit **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

Explicitly Specifying the PE Routers As Part of Virtual Ethernet Interface Configuration

There are several ways to specify the route through which traffic should pass.

- Explicitly specify the PE routers as part of the virtual Ethernet interface configuration
- Configure an MPLS Traffic Engineering tunnel
- Configure a GRE tunnel

The following task explains how to explicitly specify the PE routers as part of the virtual Ethernet interface configuration.

Note This tasks includes steps for configuring the LAN port for Layer 2 Switching. For more information, see the "Configuring LAN Ports for Layer 2 Switching." task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-ethernet** *num*
4. **transport vpls mesh**
5. **neighbor** *remote-router-id* [**pw-class** *pw-class-name*]
6. **exit**
7. **switchport**
8. **switchport mode trunk**
9. **switchport trunk allowed vlan** {**add** | **except** | **none** | **remove**} *vlan* [,*vlan*[,*vlan*[,...]]
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	**interface virtual-ethernet** *num* **Example:** `Device(config)# interface virtual-ethernet 1`	Creates a virtual Ethernet interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**transport vpls mesh** **Example:** `Device(config-if)# transport vpls mesh`	Create a full mesh of pseudowires and enters VPLS transport mode.
Step 5	**neighbor** *remote-router-id* [**pw-class** *pw-class-name*] **Example:** `Device(config-if-transport)# neighbor 10.19.19.19 pw-class 1`	Specifies the PE routers to be used in the pseudowire.
Step 6	**exit** **Example:** `Device(config-if-transport)# exit`	Exits VPLS transport configuration mode and enters interface configuration mode.
Step 7	**switchport** **Example:** `Device(config-if)# switchport`	Configures the port for Layer 2 switching.
Step 8	**switchport mode trunk** **Example:** `Device(config-if)# switchport mode trunk`	Enables permanent trunking mode and negotiates to convert the link into a trunk link.
Step 9	**switchport trunk allowed vlan** {**add** | **except** | **none** | **remove**} *vlan* [,*vlan*[,*vlan*[,...]] **Example:** `Device(config-if)# switchport trunk allowed vlan except 10, 20`	Configures the list of VLANs allowed on the trunk.
Step 10	**end** **Example:** `Device(config)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring an MPLS Traffic Engineering Tunnel

There are several ways to specify the route through which traffic should pass.

- Explicitly specify the PE devices as part of the virtual Ethernet interface configuration
- Configure an MPLS Traffic Engineering tunnel
- Configure a GRE tunnel

The following task explains how to configure an MPLS Traffic Engineering tunnel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *type number*
5. **tunnel destination** *ip-address*
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng autoroute announce**
8. **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name*} | **identifier** *path-number*} [**lockdown**]
9. **exit**
10. **ip route** *ip-address* **tunnel** *num*

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** *number* **Example:** `Device(config)# interface tunnel10`	Configures an interface type and enters interface configuration mode.
Step 4	**ip unnumbered** *type number* **Example:** `Device(config-if)# ip unnumbered loopback 0`	Assigns an IP address to the tunnel interface. • An MPLS traffic engineering tunnel interface should be unnumbered because it represents a unidirectional link.
Step 5	**tunnel destination** *ip-address*	Specifies the destination for a tunnel.

	Command or Action	Purpose
	Example: `Device(config-if)# tunnel destination 10.20.1.1`	• The *ip-address* keyword is the IP address of the host destination expressed in dotted decimal notation.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Device(config-if)# tunnel mode mpls traffic-eng`	Configures the tunnel encapsulation mode to MPLS traffic engineering.
Step 7	**tunnel mpls traffic-eng autoroute announce** **Example:** `Device(config-if)# tunnel mpls traffic-eng autoroute announce`	Configures the IGP to use the tunnel in its enhanced SPF calculation.
Step 8	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name*} | **identifier** *path-number*} [**lockdown**] **Example:** `Device(config-if)# tunnel mpls traffic-eng path-option 1 explicit name TestPath`	Configures the tunnel to use a named IP explicit path or a path dynamically calculated from the traffic engineering topology database. • A dynamic path is used if an explicit path is currently unavailable.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**ip route** *ip-address* **tunnel** *num* **Example:** `Device(config)# ip route 10.19.19.19 255.255.255.255 tunnel10`	Creates a static route.

Configuring a GRE Tunnel

There are several ways to specify the route through which traffic should pass.

- Explicitly specify the PE devices as part of the virtual Ethernet interface configuration
- Configure an MPLS Traffic Engineering tunnel
- Configure a GRE tunnel

The following task explains how to configure a GRE tunnel. For more information on GRE tunnels, see the Implementing Tunnels module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **tunnel mode** {**gre ip** | **gre multipoint**}
5. **mpls ip**
6. **tunnel source** {*ip-address* | *interface-type interface-number*}
7. **tunnel destination** {*hostname* | *ip-address*}
8. **exit**
9. **ip route** *ip-address* **tunnel** *num*

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 tunnel 1`	Specifies the interface type and number and enters interface configuration mode. • To configure a tunnel, use **tunnel** for the *type* argument.
Step 4	**tunnel mode** {**gre ip**	**gre multipoint**} **Example:** `Device(config-if)# tunnel mode gre ip`	Specifies the encapsulation protocol to be used in the tunnel.
Step 5	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables MPLS on the tunnel.
Step 6	**tunnel source** {*ip-address*	*interface-type interface-number*}	Configures the tunnel source.

	Command or Action	Purpose
	Example: `Device(config-if)# tunnel source 1.1.1.1`	• Use the *ip-address* argument to specify the source IP address. • Use the *interface-type* and *interface-number* arguments to specify the interface to use. **Note** The tunnel source and destination IP addresses must be defined on both PE Devices.
Step 7	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Device(config-if)# tunnel destination 3.3.3.3`	Configures the tunnel destination. • Use the *hostname* argument to specify the name of the host destination. • Use the *ip-address* argument to specify the IP address of the host destination. **Note** The tunnel source and destination IP addresses must be defined on both PE Devices.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**ip route** *ip-address* **tunnel** *num* **Example:** `Device(config)# ip route 10.19.19.19 255.255.255.255 Tunnel1`	Creates a static route.

Configuration Examples for L2VPN Advanced VPLS

The following sections show configuration examples for the three supported methods of configuring the L2VPN Advanced VPLS feature.

Example: Configuring L2VPN Advanced VPLS—Explicitly Specifying Peer PE Devices

The following example shows how to create two VPLS domains under VLANs 10 and 20. Each VPLS domain includes two pseudowires to peer PE devices 10.2.2.2 and 10.3.3.3. Load-balancing is enabled through the **load-balance flow** and **flow-label enable** commands.

```
pseudowire-class cl1
 encap mpls
```

```
 load-balance flow
 flow-label enable
!
port-channel load-balance src-mac
!
interface virtual-ethernet 1
 transport vpls mesh
 neighbor 10.2.2.2 pw-class cl1
 neighbor 10.3.3.3 pw-class cl1
 switchport
 switchport mode trunk
 switchport trunk allowed vlan 10, 20
```

Example: Configuring L2VPN Advanced VPLS—Using MPLS Traffic Engineering Tunnels

The following example shows the creation of two VPLS domains and uses MPLS Traffic Engineering tunnels to specify the explicit path.

```
pseudowire-class cl1
   encap mpls
!
port-channel load-balance src-mac
!
interface Tunnel1
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 192.168.1.1
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 1 explicit name LSP1
!
ip explicit-path name LSP1 enable
 next-address 192.168.2.2
 next-address loose 192.168.1.1
!
interface Tunnel2
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 172.16.1.1
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 1 explicit name LSP2
!
ip explicit-path name LSP2 enable
 next-address 172.16.2.2
 next-address loose 172.16.1.1
!
interface virtual-ethernet 1
 transport vpls mesh
 neighbor 10.2.2.2 pw-class cl1
 neighbor 10.3.3.3 pw-class cl1
 switchport
 switchport mode trunk
 switchport trunk allowed vlan 10,20

ip route 10.2.2.2 255.255.255.255 Tunnel1
ip route 10.3.3.3 255.255.255.255 Tunnel2
```

Example: Configuring L2VPN Advanced VPLS—Using MPLS over GRE Tunnels

The following example shows the creation of two VPLS domains under VLANs 10 and 20. Each VPLS domain includes two pseudowires to peer PEs 10.2.2.2 and 10.3.3.3. The pseudowires are MPLS over GRE tunnels because the core is IP.

```
pseudowire-class cl1
 encap mpls
 load-balance flow
!
port-channel load-balance src-mac
!
int tunnel 1
 tunnel mode gre ip
 mpls ip
 tunnel source 10.1.1.1
 tunnel destination 10.2.2.2
!
int tunnel 2
 tunnel mode gre ip
 mpls ip
 tunnel source 10.1.1.1
 tunnel destination 10.3.3.3
!
interface virtual-ethernet 1
 transport vpls mesh
 neighbor 10.2.2.2 pw-class cl1
 neighbor 10.3.3.3 pw-class cl1
 switchport
 switchport mode trunk
 switchport trunk allowed vlan 10, 20
ip route 10.2.2.2 255.255.255.255 Tunnel1
ip route 10.3.3.3 255.255.255.255 Tunnel2
```

Additional References for L2VPN Advanced VPLS

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
VPLS	Cisco 7600 Series Ethernet Services Plus (ES+) and Ethernet Services Plus T (ES+T) Line Card Configuration Guide
MPLS Traffic Engineering tunnels	“MPLS Traffic Engineering and Enhancements”
GRE tunnels	“Implementing Tunnels”
Cisco 6500 LAN ports	“ Configuring LAN Ports for Layer 2 Switching”

Standards

Standard	Title
draft-bryant-filsfils-fat-pw	Internet Draft:*Flow Aware Transport of MPLS Pseudowires (FAT PWs)*

RFCs

RFC	Title
RFC 4762	*Virtual Private LAN Services (VPLS) Using Label Distribution Protocol (LDP) Singling*

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

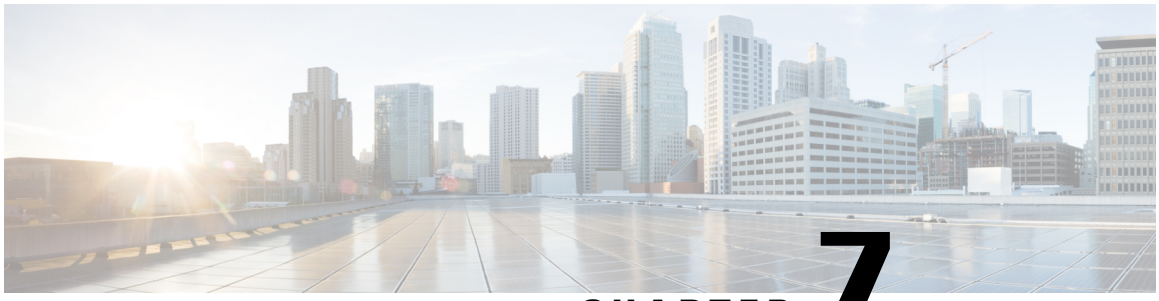
Feature Information for L2VPN Advanced VPLS

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 19: Feature Information for L2VPN Advanced VPLS

Feature Name	Releases	Feature Information
L2VPN Advanced VPLS	12.2(33)SXI4 15.1(1)SY	The L2VPN Advanced VPLS feature uses Virtual Switch System (VSS) and Flow Aware Transport (FAT) pseudowires to achieve PE redundancy and load-balancing. In 12.2(33)SXI4, this feature was introduced on the Cisco 6500 series router. The following commands were introduced: **flow-label enable**, **interfacevirtual-ethernet**, **load-balanceflow**, **neighbor (VPLS transport mode)**, **show interface virtual-ethernet**, and **transport vpls mesh**. The following command was modified: **show mpls l2transport vc**



CHAPTER 7

H-VPLS N-PE Redundancy for QinQ Access

The H-VPLS N-PE Redundancy for QinQ Access feature enables two network provider edge (N-PE) devices to provide failover services to a user provider edge (U-PE) device in a hierarchical virtual private LAN service (H-VPLS). Having redundant N-PE devices provides improved stability and reliability against link and node failures.

Finding Feature Information

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

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

Prerequisites for H-VPLS N-PE Redundancy for QinQ Access

- Before configuring this feature, configure your hierarchical virtual private LAN service (H-VPLS) network and make sure it is operating correctly.

- Make sure that the PE-to-customer edge (CE) interface is configured with a list of allowed VLANs.
- To provide faster convergence, you can enable the MPLS Traffic Engineering—Fast Reroute feature in the Multiprotocol Label Switching (MPLS) core.
- Enable the L2VPN Pseudowire Redundancy feature on the user provider edge (U-PE) devices for MPLS access.
- When configuring Multiple Spanning Tree Protocol (MSTP), specify that one of the network provider edge (N-PE) devices is the root by assigning it the lowest priority using the **spanning-tree mst** *instance-id* **priority** *priority* command.
- When configuring MSTP, make sure that each device participating in the spanning tree is in the same region and is the same revision by issuing the **revision**, **name**, and **instance** commands in MST configuration mode.

Restrictions for H-VPLS N-PE Redundancy for QinQ Access

- This feature cannot be used with the VPLS Autodiscovery feature on pseudowires that attach to network provider edge (N-PE) devices. When you create the virtual private LAN service (VPLS), you can manually create the virtual forwarding instance (VFI).
- You cannot configure more than one pseudowire to carry the bridge protocol data unit (BPDU) packets between two redundant network provider edge (N-PE) devices on the same Virtual Private LAN service (VPLS) site.
- You cannot configure a local loopback address as a neighbor when you configure the H-VPLS N-PE Redundancy feature on N-PE devices. If you do so, the following error message is displayed:

```
VPLS local switching to peer address not supported
```

- Only two N-PE devices can be connected to each U-PE device.
- The spanning-tree mode must be Multiple Spanning Tree Protocol (MSTP) for the H-VPLS N-PE Redundancy feature. If the spanning-tree mode changes, the H-VPLS N-PE Redundancy feature might not work correctly, even though the pseudowire that carries the BPDU packet still exists and the H-VPLS N-PE Redundancy feature is still configured.

Information About H-VPLS N-PE Redundancy for QinQ Access

How H-VPLS N-PE Redundancy for QinQ Access Works

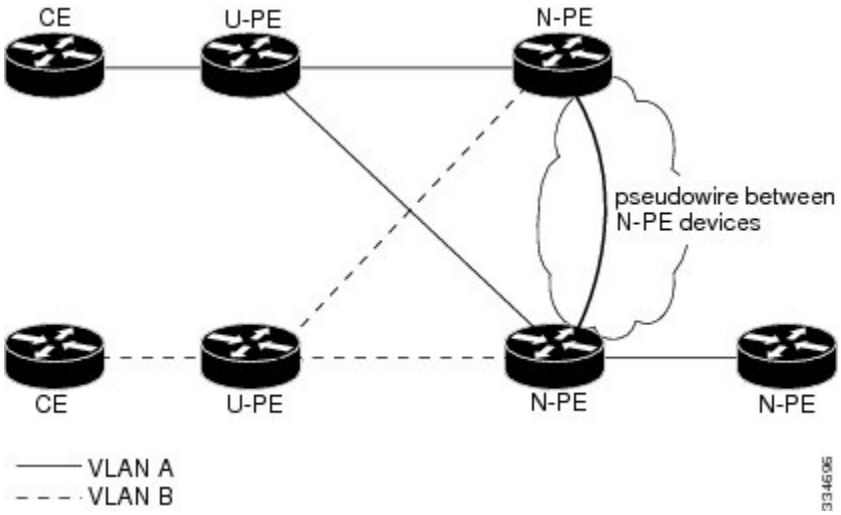
In a network configured with the H-VPLS N-PE Redundancy feature, the user provider edge (U-PE) device is connected to two network provider edge (N-PE) devices. This feature provides a level of redundancy that can tolerate both link and device faults. If a failure occurs in the network that disables one N-PE device from transmitting data, the other N-PE device takes over. This feature works with both QinQ access based on Multiple Spanning Tree Protocol (MSTP) and Multiprotocol Label Switching (MPLS) access based on pseudowire redundancy.

H-VPLS N-PE Redundancy with QinQ Access Based on MSTP

The H-VPLS N-PE Redundancy with QinQ Access feature uses the Multiple Spanning Tree Protocol (MSTP) running on the network provider edge (N-PE) devices and user provider edge (U-PE) devices in a hierarchical Virtual Private LAN service (H-VPLS) network. A pseudowire running between N-PE devices carries only MSTP bridge protocol data units (BPDUs). The pseudowire running between the N-PE devices is always up and is used to create a loop path between N-PE devices so that MSTP blocks one of the redundant paths between the U-PE device and the N-PE devices. If the primary N-PE device or the path to it fails, MSTP enables the path to the backup N-PE device.

The figure below shows an H-VPLS network with redundant access. Each U-PE device has two connections, one to each N-PE device. Between the two N-PE devices is a pseudowire to provide a loop path for MSTP BPDUs. The network topology allows for the backup N-PE device to take over if the primary N-PE device or the path to it fails.

Figure 12: H-VPLS N-PE Redundancy with QinQ Access Based on MSTP



How to Configure H-VPLS N-PE Redundancy for QinQ Access

Configuring the VPLS Pseudowire Between the N-PE Devices

Configuring network provider edge (N-PE) redundancy in a hierarchical Virtual Private LAN service (H-VPLS) network requires that you define the VPLS pseudowire for transporting bridge protocol data unit (BPDU) packets (described here) and that you connect that pseudowire to the native VLAN (described in the next task). This configuration provides a redundancy that provides improved reliability against link and node failures.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *name* **manual**
4. **vpn id** *id-number*
5. **forward permit l2protocol all**
6. **neighbor** *remote-router-id vc-id* {**encapsulation** *encapsulation-type* | **pw-class** *pw-name*} [**no-split-horizon**]
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	**l2 vfi** *name* **manual** **Example:** `Device(config)# l2 vfi vfitest1 manual`	Creates a Layer 2 virtual forwarding interface (VFI) and enters Layer 2 VFI manual configuration mode.
Step 4	**vpn id** *id-number* **Example:** `Device(config-vfi)# vpn id 200`	Specifies the VPN ID.
Step 5	**forward permit l2protocol all** **Example:** `Device(config-vfi)# forward permit l2protocol all`	Creates a pseudowire that is to be used to transport BPDU packets between the two N-PE devices.

	Command or Action	Purpose
Step 6	**neighbor** *remote-router-id vc-id* {**encapsulation** *encapsulation-type* | **pw-class** *pw-name*} [**no-split-horizon**] **Example:** `Device(config-vfi)# neighbor 10.2.2.2 3 encapsulation mpls`	Specifies the peer IP address of the redundant N-PE device and the type of tunnel signaling and encapsulation mechanism.
Step 7	**end** **Example:** `Device(config-vfi)# end`	Exits Layer 2 VFI manual configuration mode and returns to privileged EXEC mode.

Configuring the SVI for the Native VLAN

Perform this task to configure the switched virtual interface (SVI) for the native VLAN and verify that it is correctly configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface vlan** *vlan-id*
4. **xconnect vfi** *vfi-name*
5. **end**
6. **show vfi** *vfi-name*
7. **end**

DETAILED STEPS

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

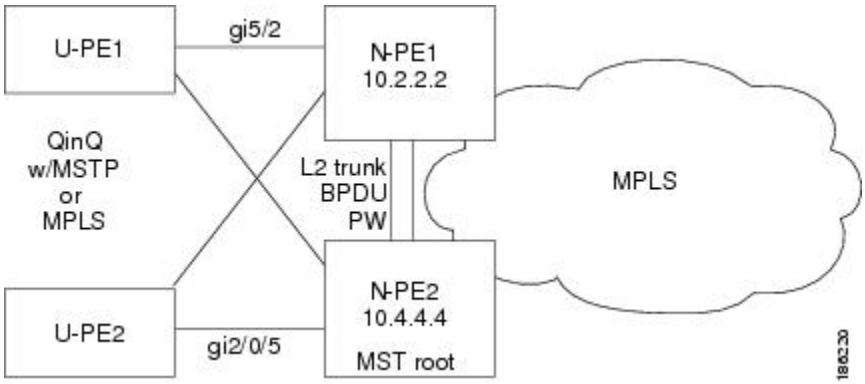
	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface vlan** *vlan-id* **Example:** `Device(config)# interface vlan 23`	Creates a dynamic SVI. • To make the SVI active when you create a VLAN, you must configure the VLAN with at least one physical interface that is in the "up" state. Use the **show vfi** command to display the status of the SVI. The state field will display "up" when the SVI is active.
Step 4	**xconnect vfi** *vfi-name* **Example:** `Device(config)# xconnect vfi vfitest1`	Specifies the Layer 2 virtual forwarding interface (VFI) that you are binding to the VLAN port.
Step 5	**end** **Example:** `Device(config-vfi)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 6	**show vfi** *vfi-name* **Example:** `Device# show vfi VPLS-2`	(Optional) Displays information about the pseudowire between the two network provider edge (N-PE) devices so that you can verify that the H-VPLS N-PE Redundancy feature is correctly configured.
Step 7	**end** **Example:** `Device# end`	Exits privileged EXEC mode and returns to user EXEC mode.

Configuration Examples for H-VPLS N-PE Redundancy for QinQ Access

Example: H-VPLS N-PE Redundancy for QinQ Access

The figure below shows a configuration that is set up for the H-VPLS N-PE Redundancy with QinQ Access feature.

Figure 13: H-VPLS N-PE Redundancy with QinQ Access Topology



The table below shows the configuration of two network provider edge (N-PE) devices.

Table 20: Example: H-VPLS N-PE Redundancy for QinQ Access

N-PE1	N-PE2

`l2 vfi l2trunk manual`
` vpn id 10`
` forward permit l2protocol all`
` neighbor 10.4.4.4 encapsulation mpls`
`!`
`interface Vlan1`
` no ip address`
` xconnect vfi l2trunk`
`!`
`spanning-tree mode mst`
`spanning-tree extend system-id`
`!`
`spanning-tree mst configuration`
` revision 10`
` instance 1 vlan 20`
`!`
`interface GigabitEthernet5/2`
` switchport`
` switchport trunk encapsulation dot1q`
` switchport trunk allowed vlan 20`
` switchport mode trunk`
	`l2 vfi l2trunk manual`
	` vpn id 10`
	` forward permit l2protocol all`
	` neighbor 10.2.2.2 encapsulation mpls`
	`!`
	`interface Vlan1`
	` no ip address`
	` xconnect vfi l2trunk`
	`!`
	`spanning-tree mode mst`
	`spanning-tree extend system-id`
	`!`
	`spanning-tree mst configuration`
	` revision 10`
	` instance 1 vlan 20`
	`!`
	`spanning-tree mst 1 priority 0`
	`!`
	`interface GigabitEthernet2/0/5`
	` switchport`
	` switchport trunk allowed vlan 20`
	` switchport mode trunk`
	` mls qos trust dscp`

Additional References

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
L2VPN pseudowire redundancy	"L2VPN Pseudowire Redundancy" feature module in the *MPLS Layer 2 VPNs Configuration Guide*.
H-VPLS	"Configuring VPLS" in the "Configuring Multiprotocol Label Switching on the Optical Services Modules" chapter in the *Optical Services Modules Installation and Configuration Notes*, 12.2SR document.
MPLS traffic engineering	"MPLS Traffic Engineering Fast Reroute Link and Node Protection" feature module in the *MPLS Traffic Engineering: Path, Link, and Node Protection Configuration Guide* (part of the Multiprotocol Label Switching Configuration Guide Library)

Standards

Standard	Title
http://www.ietf.org/rfc/rfc4447.txt	*Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*
http://www3.ietf.org/proceedings/06mar/IDs/draft-ietf-l2vpn-vpls-ldp-08.txt	*Virtual Private LAN Services over MPLS*
http://www.ietf.org/internet-drafts/draft-ietf-pwe3-segmented-pw-02.txt	*Segmented Pseudo Wire*
draft-ietf-pwe3-vccv-10.txt	*Pseudo Wire Virtual Circuit Connectivity Verification (VCCV)*
draft-ietf-pwe3-oam-msg-map-03.txt	*Pseudo Wire (PW) OAM Message Mapping*

MIBs

MIB	MIBs Link
Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services	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

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 H-VPLS N-PE Redundancy for QinQ Access

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 21: Feature Information for H-VPLS N-PE Redundancy for QinQ Access

Feature Name	Releases	Feature Information
H-VPLS N-PE Redundancy for QinQ Access	12.2(33)SRC 12.2(50)SY Cisco IOS XE Release 3.8S	The H-VPLS N-PE Redundancy for QinQ Access feature provides the capability to dual-home a given user provider edge (U-PE) device to two network provide edge (N-PE) devices in order to provide protection against link and node failures. In Cisco IOS Release 12.2(33)SRC, this feature was introduced on the Cisco 7600 series routers. In Cisco IOS Release 12.2(50)SY, this feature was integrated. In Cisco IOS XE Release 3.8S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **forward permit l2protocol**, **show mpls l2transport vc**.

Glossary

CE device—customer edge device. A device that belongs to a customer network, which connects to a PE device to utilize MPLS VPN network services.

LAN—local-area network. High-speed, low-error data network covering a relatively small geographic area. LANs connect workstations, peripherals, terminals, and other devices in a single building or other geographically limited areas.

MPLS—Multiprotocol Label Switching. A packet-forwarding technology, used in the network core, that applies data link layer labels to tell switching nodes how to forward data, resulting in faster and more scalable forwarding than network layer routing normally can do.

MSTP—Multiple Spanning Tree Protocol. MSTP enables multiple VLANs to be mapped to the same spanning-tree instance, reducing the number of spanning-tree instances needed to support a large number of VLANs.

N-PE—network provider edge device. This device acts as a gateway between the MPLS core and edge domains.

PE device—provider edge device. The PE device is the entry point into the service provider network. The PE device is typically deployed on the edge of the network and is administered by the service provider.

pseudowire—A pseudowire is a virtual connection that, in the context of VPLS, connects two SVIs. It is a mechanism that carries the elements of an emulated service from one PE device to one or more PE devices over a packet switched network (PSN). A pseudowire is bidirectional and consists of a pair of unidirectional MPLS virtual circuits (VCs). A pseudowire can be used to connect a point-to-point circuit.

QinQ—An IEEE 802.1Q VLAN tunnel. A mechanism for constructing multipoint Layer 2 VPN using Ethernet switches.

redundancy—The duplication of devices, services, or connections so that, in the event of a failure, they can perform the work of those that failed.

router—A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

spanning tree—Loop-free subset of a network topology.

U-PE—user provider edge device. This device connects CE devices to the service.

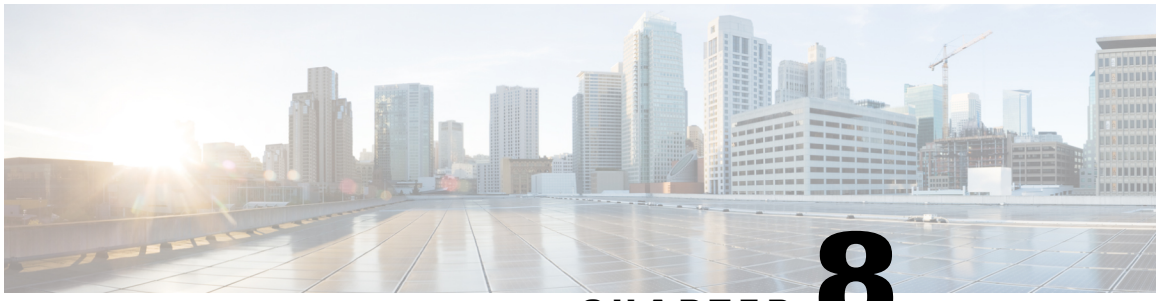
VFI—virtual forwarding instance. A VFI is a collection of data structures used by the data plane, software-based or hardware-based, to forward packets to one or more VCs.

VLAN—Virtual LAN. Group of devices on one or more LANs that are configured (using management software) so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments.

VPLS—Virtual Private LAN Service. VPLS describes an architecture that delivers Layer 2 service that emulates an Ethernet LAN across a wide-area network (WAN) and inherits the scaling characteristics of a LAN.

VPLS redundancy—Also called N-PE redundancy. Allows U-PEs to be dual-honed (to their N-PEs) in a loop-free topology with MPLS or QinQ as the access or aggregation domain.

VPN—Virtual Private Network. Allows IP traffic to travel securely over public TCP/IP networks and the Internet by encapsulating and encrypting all IP packets. VPN uses a tunnel to encrypt all information at the IP level.



CHAPTER 8

H-VPLS N-PE Redundancy for MPLS Access

The H-VPLS N-PE Redundancy for MPLS Access feature enables two network provider edge (N-PE) devices to provide failover services to a user provider edge (U-PE) device in a hierarchical virtual private LAN service (H-VPLS). Having redundant N-PE devices provides improved stability and reliability against link and node failures.

Finding Feature Information

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

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

Prerequisites for H-VPLS N-PE Redundancy for MPLS Access

- Before configuring this feature, configure your hierarchical virtual private LAN service (H-VPLS) network and make sure it is operating correctly.

- Make sure that the PE-to-customer edge (CE) interface is configured with a list of allowed VLANs.
- To provide faster convergence, you can enable the MPLS Traffic Engineering—Fast Reroute feature in the Multiprotocol Label Switching (MPLS) core.
- Enable the L2VPN Pseudowire Redundancy feature on the user provider edge (U-PE) devices for MPLS access.

Restrictions for H-VPLS N-PE Redundancy for MPLS Access

- This feature cannot be used with the VPLS Autodiscovery feature on pseudowires that attach to user provider edge (U-PE) devices. When you create the virtual private LAN service (VPLS), you can manually create the virtual forwarding interface (VFI).
- You cannot configure more than one pseudowire to carry the bridge protocol data unit (BPDU) information between the network provider edge (N-PE) devices.
- You cannot configure a local loopback address as a neighbor when you configure the H-VPLS N-PE Redundancy feature on N-PE devices.
- Only two N-PE devices can be connected to each U-PE device.

Information About H-VPLS N-PE Redundancy for MPLS Access

How H-VPLS N-PE Redundancy for MPLS Access

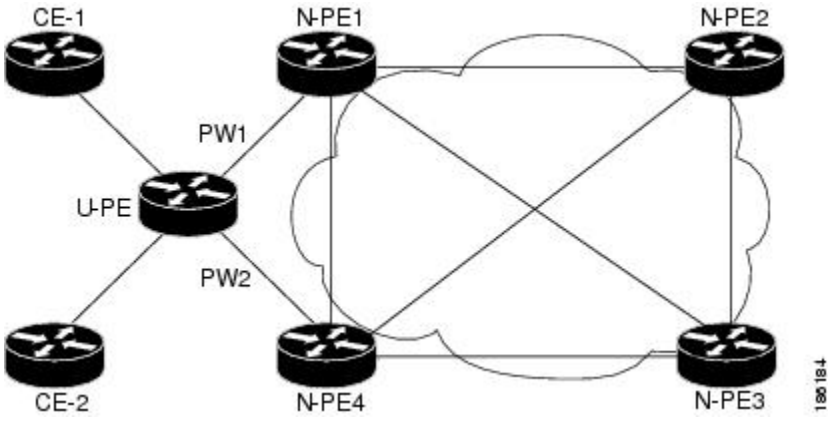
In a network configured with the H-VPLS N-PE Redundancy feature, the user provider edge (U-PE) device is connected to two network provider edge (N-PE) devices. This feature provides a level of redundancy that can tolerate both link and device faults. If a failure occurs in the network that disables one N-PE device from transmitting data, the other N-PE device takes over.

H-VPLS N-PE Redundancy with MPLS Access Based on Pseudowire Redundancy

For the H-VPLS Redundancy with MPLS Access feature based on pseudowire redundancy, the Multiprotocol Label Switching (MPLS) network has pseudowires to the virtual private LAN service (VPLS) core network provider edge (N-PE) devices.

As shown in the figure below, one pseudowire transports data between the user provider edge (U-PE) device and its peer N-PE devices. When a failure occurs along the path of the U-PE device, the backup pseudowire and the redundant N-PE device become active and start transporting data.

Figure 14: H-VPLS N-PE Redundancy for MPLS Access Based on Pseudowire Redundancy



How to Configure H-VPLS N-PE Redundancy for MPLS Access

Configuring the VPLS Pseudowire Between the N-PE Devices

Configuring network provider edge (N-PE) redundancy in a hierarchical Virtual Private LAN service (H-VPLS) network requires that you define the VPLS pseudowire for transporting bridge protocol data unit (BPDU) packets (described here) and that you connect that pseudowire to the native VLAN (described in the next task). This configuration provides a redundancy that provides improved reliability against link and node failures.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *name* **manual**
4. **vpn id** *id-number*
5. **forward permit l2protocol all**
6. **neighbor** *remote-router-id vc-id* {**encapsulation** *encapsulation-type* | **pw-class** *pw-name*} [**no-split-horizon**]
7. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2 vfi** *name* **manual** **Example:** `Device(config)# l2 vfi vfitest1 manual`	Creates a Layer 2 virtual forwarding interface (VFI) and enters Layer 2 VFI manual configuration mode.
Step 4	**vpn id** *id-number* **Example:** `Device(config-vfi)# vpn id 200`	Specifies the VPN ID.
Step 5	**forward permit l2protocol all** **Example:** `Device(config-vfi)# forward permit l2protocol all`	Creates a pseudowire that is to be used to transport BPDU packets between the two N-PE devices.
Step 6	**neighbor** *remote-router-id vc-id* {**encapsulation** *encapsulation-type* | **pw-class** *pw-name*} [**no-split-horizon**] **Example:** `Device(config-vfi)# neighbor 10.2.2.2 3 encapsulation mpls`	Specifies the peer IP address of the redundant N-PE device and the type of tunnel signaling and encapsulation mechanism.
Step 7	**end** **Example:** `Device(config-vfi)# end`	Exits Layer 2 VFI manual configuration mode and returns to privileged EXEC mode.

Configuring the SVI for the Native VLAN

Perform this task to configure the switched virtual interface (SVI) for the native VLAN and verify that it is correctly configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface vlan** *vlan-id*
4. **xconnect vfi** *vfi-name*
5. **end**
6. **show vfi** *vfi-name*
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 vlan** *vlan-id* **Example:** `Device(config)# interface vlan 23`	Creates a dynamic SVI. • To make the SVI active when you create a VLAN, you must configure the VLAN with at least one physical interface that is in the "up" state. Use the **show vfi** command to display the status of the SVI. The state field will display "up" when the SVI is active.
Step 4	**xconnect vfi** *vfi-name* **Example:** `Device(config)# xconnect vfi vfitest1`	Specifies the Layer 2 virtual forwarding interface (VFI) that you are binding to the VLAN port.
Step 5	**end** **Example:** `Device(config-vfi)# end`	Ends the current configuration session and returns to privileged EXEC mode.

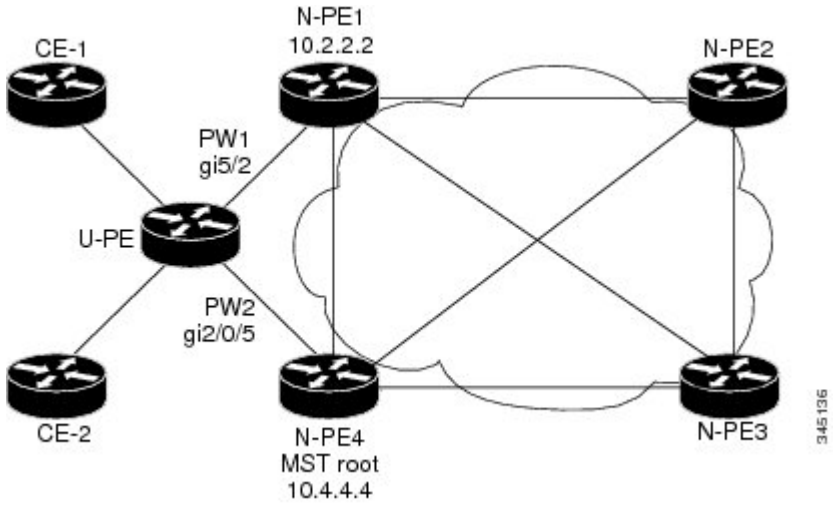
	Command or Action	Purpose
Step 6	**show vfi** *vfi-name* **Example:** `Device# show vfi VPLS-2`	(Optional) Displays information about the pseudowire between the two network provider edge (N-PE) devices so that you can verify that the H-VPLS N-PE Redundancy feature is correctly configured.
Step 7	**end** **Example:** `Device# end`	Exits privileged EXEC mode and returns to user EXEC mode.

Configuration Examples for H-VPLS N-PE Redundancy for MPLS Access

Example: H-VPLS N-PE Redundancy for MPLS Access

The figure below shows a configuration that is set up for the H-VPLS N-PE Redundancy with MPLS Access feature.

Figure 15: H-VPLS N-PE Redundancy with MPLS Access Topology



The table below shows the configuration of two network provider edge (N-PE) devices.

Table 22: Example: H-VPLS N-PE Redundancy for MPLS Access

N-PE1	N-PE4
l2 vfi l2trunk manual vpn id 10 forward permit l2protocol all neighbor 10.4.4.4 encapsulation mpls ! interface Vlan1 no ip address xconnect vfi l2trunk ! spanning-tree mode mst spanning-tree extend system-id ! spanning-tree mst configuration revision 10 instance 1 vlan 20 ! interface GigabitEthernet5/2 switchport switchport trunk encapsulation dot1q switchport trunk allowed vlan 20 switchport mode trunk	l2 vfi l2trunk manual vpn id 10 forward permit l2protocol all neighbor 10.2.2.2 encapsulation mpls ! interface Vlan1 no ip address xconnect vfi l2trunk ! spanning-tree mode mst spanning-tree extend system-id ! spanning-tree mst configuration revision 10 instance 1 vlan 20 ! spanning-tree mst 1 priority 0 ! interface GigabitEthernet2/0/5 switchport switchport trunk allowed vlan 20 switchport mode trunk mls qos trust dscp

Additional References

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
L2VPN pseudowire redundancy	"L2VPN Pseudowire Redundancy" feature module in the *MPLS Layer 2 VPNs Configuration Guide*.
H-VPLS	"Configuring VPLS" in the "Configuring Multiprotocol Label Switching on the Optical Services Modules" chapter in the *Optical Services Modules Installation and Configuration Notes*, 12.2SR document.
MPLS traffic engineering	"MPLS Traffic Engineering Fast Reroute Link and Node Protection" feature module in the *MPLS Traffic Engineering: Path, Link, and Node Protection Configuration Guide* (part of the Multiprotocol Label Switching Configuration Guide Library)

Standards

Standard	Title
http://www.ietf.org/rfc/rfc4447.txt	*Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*
http://www3.ietf.org/proceedings/06mar/IDs/draft-ietf-l2vpn-vpls-ldp-08.txt	*Virtual Private LAN Services over MPLS*
http://www.ietf.org/internet-drafts/draft-ietf-pwe3-segmented-pw-02.txt	*Segmented Pseudo Wire*
draft-ietf-pwe3-vccv-10.txt	*Pseudo Wire Virtual Circuit Connectivity Verification (VCCV)*
draft-ietf-pwe3-oam-msg-map-03.txt	*Pseudo Wire (PW) OAM Message Mapping*

MIBs

MIB	MIBs Link
Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services	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

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 H-VPLS N-PE Redundancy for MPLS Access

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 23: Feature Information for H-VPLS N-PE Redundancy for MPLS Access

Feature Name	Releases	Feature Information
H-VPLS N-PE Redundancy for MPLS Access	Cisco IOS XE Release 3.6S	The H-VPLS N-PE Redundancy for MPLS Access feature enables two network provider edge (N-PE) devices to provide redundancy to a user provider edge (U-PE) device in a hierarchical virtual private LAN service (H-VPLS). Having redundant N-PE devices provides improved stability and reliability against link and node failures. In Cisco IOS XE Release 3.6S, support was added for the Cisco ASR 903 Router. The following commands were introduced or modified: **forward permit l2protocol**, **show mpls l2transport vc**.

Glossary

CE device—customer edge device. A device that belongs to a customer network, which connects to a PE device to utilize MPLS VPN network services.

LAN—local-area network. High-speed, low-error data network covering a relatively small geographic area. LANs connect workstations, peripherals, terminals, and other devices in a single building or other geographically limited areas.

MPLS—Multiprotocol Label Switching. A packet-forwarding technology, used in the network core, that applies data link layer labels to tell switching nodes how to forward data, resulting in faster and more scalable forwarding than network layer routing normally can do.

MSTP—Multiple Spanning Tree Protocol. MSTP enables multiple VLANs to be mapped to the same spanning-tree instance, reducing the number of spanning-tree instances needed to support a large number of VLANs.

N-PE—network provider edge device. This device acts as a gateway between the MPLS core and edge domains.

PE device—provider edge device. The PE device is the entry point into the service provider network. The PE device is typically deployed on the edge of the network and is administered by the service provider.

pseudowire—A pseudowire is a virtual connection that, in the context of VPLS, connects two SVIs. It is a mechanism that carries the elements of an emulated service from one PE device to one or more PE devices over a packet switched network (PSN). A pseudowire is bidirectional and consists of a pair of unidirectional MPLS virtual circuits (VCs). A pseudowire can be used to connect a point-to-point circuit.

QinQ—An IEEE 802.1Q VLAN tunnel. A mechanism for constructing multipoint Layer 2 VPN using Ethernet switches.

redundancy—The duplication of devices, services, or connections so that, in the event of a failure, they can perform the work of those that failed.

router—A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

spanning tree—Loop-free subset of a network topology.

U-PE—user provider edge device. This device connects CE devices to the service.

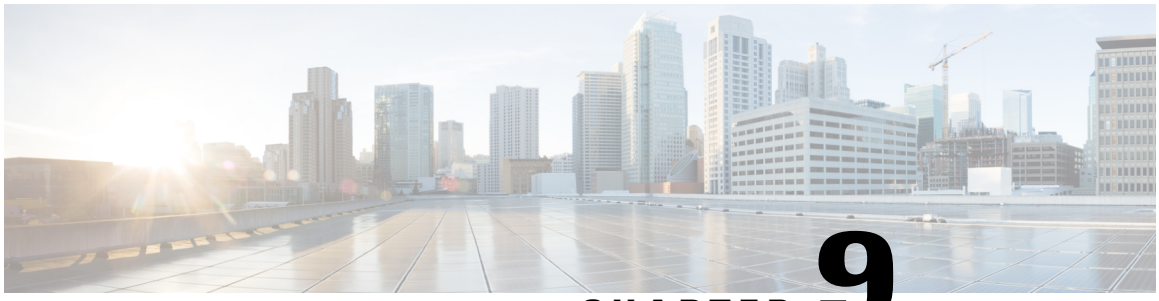
VFI—virtual forwarding instance. A VFI is a collection of data structures used by the data plane, software-based or hardware-based, to forward packets to one or more VCs.

VLAN—Virtual LAN. Group of devices on one or more LANs that are configured (using management software) so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments.

VPLS—Virtual Private LAN Service. VPLS describes an architecture that delivers Layer 2 service that emulates an Ethernet LAN across a wide-area network (WAN) and inherits the scaling characteristics of a LAN.

VPLS redundancy—Also called N-PE redundancy. Allows U-PEs to be dual-honed (to their N-PEs) in a loop-free topology with MPLS or QinQ as the access or aggregation domain.

VPN—Virtual Private Network. Allows IP traffic to travel securely over public TCP/IP networks and the Internet by encapsulating and encrypting all IP packets. VPN uses a tunnel to encrypt all information at the IP level.



CHAPTER 9

VPLS MAC Address Withdrawal

The VPLS MAC Address Withdrawal feature provides faster convergence by removing (or unlearning) MAC addresses that have been dynamically learned. A Label Distribution Protocol (LDP)-based MAC address withdrawal message is used for this purpose. A MAC list Type Length Value (TLV) is part of the MAC address withdrawal message. No configuration is needed.

Finding Feature Information

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

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

Information About VPLS MAC Address Withdrawal

VPLS MAC Address Withdrawal

The VPLS MAC Address Withdrawal feature provides faster convergence by removing (or unlearning) MAC addresses that have been dynamically learned. A Label Distribution Protocol (LDP)-based MAC address withdrawal message is used for this purpose. A MAC list Type Length Value (TLV) is part of the MAC address withdrawal message.

The **debug mpls ldp messages** and **debug mpls ldp session io** commands support monitoring of MAC address withdrawal messages being exchanged between LDP peers. Any Transport over Multiprotocol Label Switching

(AToM) might provide other means to display or monitor MAC address withdrawal messages. The Tag Distribution Protocol (TDP) is not supported because AToM uses only LDP for the MAC address withdrawal message.

PE devices learn the remote MAC addresses and directly attached MAC addresses on customer-facing ports by deriving the topology and forwarding information from packets originating at customer sites. To display the number of MAC address withdrawal messages, enter the **show mpls l2transport vc detail** command, as shown in the following example:

```
Device# show mpls l2transport vc detail

Local interface: VFI TEST VFI up
  MPLS VC type is VFI, interworking type is Ethernet
  Destination address: 10.1.1.1, VC ID: 1000, VC status: up
    Output interface: Se2/0, imposed label stack {17}
    Preferred path: not configured
    Default path: active
    Next hop: point2point
  Create time: 00:04:34, last status change time: 00:04:15
  Signaling protocol: LDP, peer 10.1.1.1:0 up
    Targeted Hello: 10.1.1.1(LDP Id) -> 10.1.1.1
    MPLS VC labels: local 16, remote 17
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
    MAC Withdraw: sent 5, received 3
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 0, send 0
    byte totals:   receive 0, send 0
    packet drops:  receive 0, send 0
```

VPLS MAC Address Withdrawal using the commands associated with the L2VPN Protocol-Based CLIs feature

The VPLS MAC Address Withdrawal feature provides faster convergence by removing (or unlearning) MAC addresses that have been dynamically learned. A Label Distribution Protocol (LDP)-based MAC address withdrawal message is used for this purpose. A MAC list Type Length Value (TLV) is part of the MAC address withdrawal message.

The **debug mpls ldp messages** and **debug mpls ldp session io** commands support monitoring of MAC address withdrawal messages being exchanged between LDP peers. Any Transport over Multiprotocol Label Switching (AToM) might provide other means to display or monitor MAC address withdrawal messages. The Tag Distribution Protocol (TDP) is not supported because AToM uses only LDP for the MAC address withdrawal message.

PE devices learn the remote MAC addresses and directly attached MAC addresses on customer-facing ports by deriving the topology and forwarding information from packets originating at customer sites. To display the number of MAC address withdrawal messages, enter the **show l2vpn atom vc detail** command, as shown in the following example:

```
Device# show l2vpn atom vc detail

Local interface: VFI TEST VFI up
  MPLS VC type is VFI, interworking type is Ethernet
  Destination address: 10.1.1.1, VC ID: 1000, VC status: up
    Output interface: Se2/0, imposed label stack {17}
    Preferred path: not configured
    Default path: active
    Next hop: point2point
```

```
Create time: 00:04:34, last status change time: 00:04:15
Signaling protocol: LDP, peer 10.1.1.1:0 up
  Targeted Hello: 10.1.1.1(LDP Id) -> 10.1.1.1
  MPLS VC labels: local 16, remote 17
  Group ID: local 0, remote 0
  MTU: local 1500, remote 1500
  Remote interface description:
  MAC Withdraw: sent 5, received 3
Sequencing: receive disabled, send disabled
VC statistics:
  packet totals: receive 0, send 0
  byte totals:   receive 0, send 0
  packet drops:  receive 0, send 0
```

How MAC Address Withdrawal Works with H-VPLS N-PE Redundancy with MPLS Access

If the pseudowire between the user provider edge (U-PE) device and network provider edge (N-PE) device fails, the L2VPN Pseudowire Redundancy feature on the U-PE device activates the standby pseudowire. In addition, the U-PE device sends a Label Distribution Protocol (LDP) MAC address withdrawal request to the new N-PE device, which forwards the message to all pseudowires in the virtual private LAN service (VPLS) core and flushes its MAC address table.

If a switched virtual interface (SVI) on the N-PE device fails, the L2VPN Pseudowire Redundancy feature activates the standby pseudowire and the U-PE device sends a MAC withdrawal message to the newly active N-PE device.

How MAC Address Withdrawal Works with H-VPLS N-PE Redundancy with QinQ Access

If a failure occurs in the customer-switched network, a spanning-tree Topology Change Notification (TCN) is issued to the network provider edge (N-PE) device, which issues a Label Distribution Protocol (LDP)-based MAC address withdrawal message to the peer N-PE devices and flushes its MAC address table.

Additional References for Any Transport over MPLS

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

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

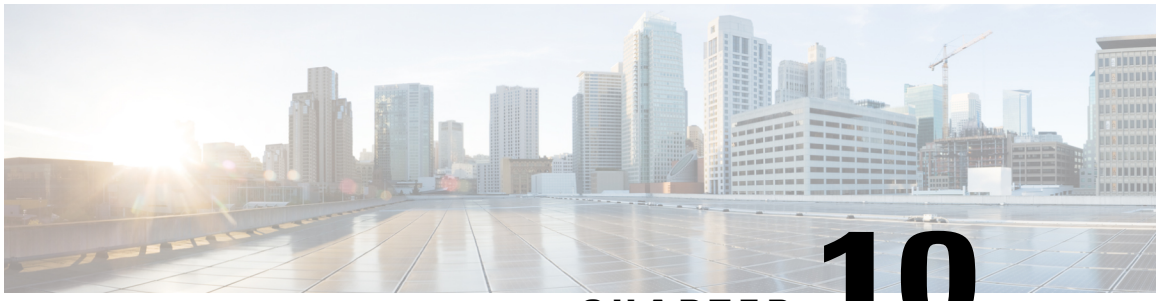
Feature Information for VPLS MAC Address Withdrawal

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 24: Feature Information for VPLS MAC Address Withdrawal

Feature Name	Releases	Feature Information
VPLS MAC Address Withdrawal	Cisco IOS XE Release 3.5S	The VPLS MAC Address Withdrawal feature provides faster convergence by removing (or unlearning) MAC addresses that have been dynamically learned. In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router. No commands were introduced or modified.



CHAPTER 10

Routed Pseudo-Wire and Routed VPLS

This feature module explains how to configure Routed Pseudo-Wire and Routed VPLS .

Finding Feature Information

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

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

Configuring Routed Pseudo-Wire and Routed VPLS

RPW and Routed VPLS can route Layer 3 traffic as well as switch Layer 2 frames for pseudowire connections between provider edge (PE) devices. Both point-to-point PE connections, in the form of Ethernet over MPLS (EoMPLS), and Virtual Private LAN Services (VPLS) multipoint PE connections are supported. The ability to route frames to and from these interfaces supports termination of a pseudowire into a Layer 3 network (VPN or global) on the same switch, or to tunnel Layer 3 frames over a Layer 2 tunnel (EoMPLS or VPLS). The feature supports faster network convergence in the event of a physical interface or device failure through the MPLS Traffic Engineering (MPLS-TE) and Fast Reroute (FRR) features. In particular, the feature enables MPLS TE-FRR protection for Layer 3 multicast over a VPLS domain.

When the RPW is configured in A-VPLS mode, TE/FRR is not supported because A-VPLS runs over ECMP and the ECMP convergence is comparable to TE/FRR.

To configure routing support for the pseudowire, configure an IP address and other Layer 3 features for the Layer 3 domain (VPN or global) in the virtual LAN (VLAN) interface configuration. The following example

assigns the IP address 10.10.10.1 to the VLAN 100 interface, and enables Multicast PIM. (Layer 2 forwarding is defined by the VFI VFI100.)

```
interface bdi 100

  ip address 10.10.10.1 255.255.255.0
```

The following example assigns an IP address 20.20.20.1 of the VPN domain VFI200. (Layer 2 forwarding is defined by the VFI VFI200.)

```
interface bdi 200

  ip address 20.20.20.1 255.255.255.0
```

Verifying Routed Pseudo-Wire and Routed VPLS Configuration

You can use the **show mpls platform** command to view information about a routed pseudowire and routed VPLS configuration.

The following example shows how to display information about a routed pseudowire and routed VPLS configuration:

SUMMARY STEPS

1. show mpls platform vpls 100

DETAILED STEPS

show mpls platform vpls 100

Example:

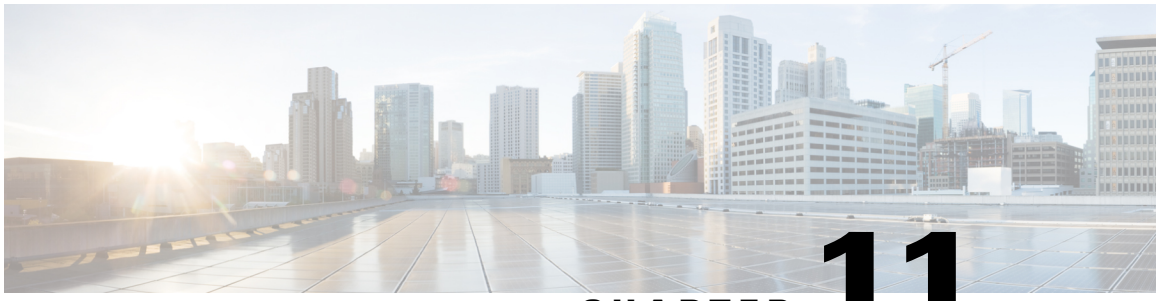
```
Device# show mpls platform vpls 100

-----------------------------
VPLS VLAN 100 (BD 100): V4
  VC info (#spoke VCs 0) :
    Imp:  tcam 224    (68    ) adj 131076 (0x20004) [peer 1.1.1.1 ID vc_id 100 2:1]  \
stats 0/0 0/0
    Disp: tcam 324    (66    ) adj 114692 (0x1C004) [in_label 16] stats 0/0
-------------------------------
BD Flood Manager: VLAN/BD 100, 3 peers, V4
  CMET handle 0x8 top 8 (0x8) bottom 3280 (0xCD0)
  Ingr flood: tcam 64/0x40 (sw 15) adj 196608   (0x30000) elif 0x701C0064 stats 0/0  \
0/0
  Egr  flood: tcam 65/0x41 (sw 72) adj 180228   (0x2C004) elif 0x701C0064 stats 0/0  \
0/0
    BD ports:       adj 32868    (0x8064) elif 0x20000064 stats 3/208
  Ingr local: tcam 32/0x20 (sw 13) adj 180224   (0x2C000) elif 0x20000064 stats 0/0
  Egr  local: tcam 33/0x21 (sw 14) adj 180225   (0x2C001) elif 0x20000064 stats 0/0
  IRB Ingr V4 Mcast control 162/0xA2 (sw 79), adj 196609   (0x30001)
      Egr  V4 Mcast control 164/0xA4 (sw 84), adj 180229   (0x2C005)
      Ingr V4 Mcast data 192/0xC0 (sw 80), adj 1966
(0x30000)
      Egr  V4 Mcast data 194/0xC2 (sw 85), adj 180228   (0x2C004)
      Ingr V4 Bcast 34/0x22 (sw 81), adj 196609   (0x30001)
      Egr  V4 Bcast 35/0x23 (sw 86), adj 180229   (0x2C005)
  IRB Ingr V6 Mcast control 608/0x260 (sw 82), adj 196608   (0x30000)
```

```
      Egr  V6 Mcast control 612/0x264 (sw 89), adj 180228   (0x2C004)
      Ingr V6 Mcast data 672/0x2A0 (sw 83), adj 196608   (0x30000)
      Egr  V6 Mcast data 676/0x2A4 (sw 90), adj 180228   (0x2C004)
      ip2irb local 36/0x24 (sw 87), adj 180226   (0x2C002) stats 0/0
      ip2irb flood 66/0x42 (sw 88), adj 180230   (0x2C006) stats 0/0
BD Flood Manager: 1 BDs, LTL base 0x90E, LTL clients: VPLS
                : Wildcard entry tcam 288 (12) adj 78089 (0x13109)
```

Feature Information for Routed Pseudo-Wire and Routed VPLS

Table 25: Feature Information for Routed Pseudo-Wire and Routed VPLS

Feature Name	Releases	Feature Information
Routed Pseudo-Wire and Routed VPLS	12.2(33)SRB 12.2(33)SXJ1 15.0(1)SY 15.2(4)M Cisco IOS XE Release 3.6S	This feature routes Layer 3 traffic as well as switch Layer 2 frames for pseudowire connections between provider edge (PE) devices. In Cisco IOS Release 12.2(33)SRB, this feature was introduced on the Cisco 7600 series routers. In Cisco IOS Release 12.2(33)SXJ1, this feature was integrated. This feature is supported on WAN cards. The following command was modified: **show mpls platform** In Cisco IOS Release 15.0(1)SY, this feature was integrated. In Cisco IOS Release 15.2(4)M, this feature was integrated. In Cisco IOS XE Release 3.6S, support was added for the Cisco ASR 1000 Series Routers.



CHAPTER 11

VPLS Autodiscovery BGP Based

VPLS Autodiscovery enables Virtual Private LAN Service (VPLS) provider edge (PE) devices to discover other PE devices that are part of the same VPLS domain. VPLS Autodiscovery also automatically detects when PE devices are added to or removed from a VPLS domain. As a result, with VPLS Autodiscovery enabled, you no longer need to manually configure a VPLS domain and maintain the configuration when a PE device is added or deleted. VPLS Autodiscovery uses the Border Gateway Protocol (BGP) to discover VPLS members and set up and tear down pseudowires in a VPLS domain.

This module describes how to configure BGP-based VPLS Autodiscovery.

Feature Information for

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 26:

Feature Name	Releases	Feature Information

Prerequisites for VPLS Autodiscovery BGP Based

Before configuring VPLS Autodiscovery, if you are using a Cisco 7600 series router, perform the Cisco 7600 router-specific tasks listed in the section called "Virtual Private LAN Services on the Optical Service Modules" in the Cisco 7600 Series Router IOS Software Configuration Guide.

Restrictions for VPLS Autodiscovery BGP Based

- Virtual Private LAN Service (VPLS) Autodiscovery supports only IPv4 addresses.
- VPLS Autodiscovery uses Forwarding Equivalence Class (FEC) 129 to convey endpoint information. Manually configured pseudowires use FEC 128.
- VPLS Autodiscovery is not supported with Layer 2 Tunnel Protocol Version 3 (L2TPv3).
- You can configure both autodiscovered and manually configured pseudowires in a single virtual forwarding instance (VFI). However, you cannot configure different pseudowires on the same peer PE device.
- After enabling VPLS Autodiscovery, if you manually configure a neighbor by using the **neighbor** command and both peers are in autodiscovery mode, each peer will receive discovery data for that VPLS. To prevent peers from receiving data for the VPLS domain, manually configure route target (RT) values.
- If you manually configure multiple pseudowires and target different IP addresses on the same PE device for each pseudowire, do not use the same virtual circuit (VC) ID to identify pseudowires that terminate at the same PE device.
- If you manually configure a neighbor on one PE device, you cannot configure the same pseudowire in the other direction by using autodiscovery on another PE device.
- Tunnel selection is not supported with autodiscovered neighbors.
- Up to 16 RTs are supported per VFI.
- The same RT is not allowed in multiple VFIs on the same PE device.
- The Border Gateway Protocol (BGP) autodiscovery process does not support dynamic, hierarchical VPLS. User-facing PE (U-PE) devices cannot discover network-facing PE (N-PE) devices, and N-PE devices cannot discover U-PE devices.
- Pseudowires for autodiscovered neighbors have split horizon enabled. (A split horizon is enabled by default on all interfaces. A split horizon blocks route information from being advertised by a device, irrespective of the interface from which the information originates.) Therefore, manually configure pseudowires for hierarchical VPLS. Ensure that U-PE devices do not participate in BGP autodiscovery for these pseudowires.
- Do not disable split horizon on autodiscovered neighbors. Split horizon is required with VPLS Autodiscovery.
- The provisioned peer address must be a /32 address bound to the peer's Label Distribution Protocol (LDP) router ID.
- A peer PE device must be able to access the IP address that is used as the local LDP router ID. Even if the IP address is not used in the **xconnect** command on the peer PE device, the IP address must be reachable.

Information About VPLS Autodiscovery BGP Based

How VPLS Works

Virtual Private LAN Service (VPLS) allows Multiprotocol Label Switching (MPLS) networks to provide multipoint Ethernet LAN services, also known as Transparent LAN Services (TLS). All customer sites in a VPLS appear to be on the same LAN, even though these sites might be in different geographic locations.

How the VPLS Autodiscovery BGP Based Feature Works

VPLS Autodiscovery enables each Virtual Private LAN Service (VPLS) provider edge (PE) device to discover other PE devices that are part of the same VPLS domain. VPLS Autodiscovery also tracks PE devices when they are added to or removed from a VPLS domain. Autodiscovery and signaling functions use the Border Gateway Protocol (BGP) to find and track PE devices.

BGP uses the Layer 2 VPN (L2VPN) Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 virtual forwarding instance (VFI) is configured. The prefix and path information is stored in the L2VPN database, which allows BGP to make decisions about the best path. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, this endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.

The BGP autodiscovery mechanism facilitates the configuration of L2VPN services, which are an integral part of the VPLS feature. VPLS enables flexibility in deploying services by connecting geographically dispersed sites as a large LAN over high-speed Ethernet in a robust and scalable IP Multiprotocol Label Switching (MPLS) network. For more information about BGP and the L2VPN address family in relation to VPLS Autodiscovery, see the following chapters in the *IP Routing: BGP Configuration Guide*:

- “L2VPN Address Family” section in the “Cisco BGP Overview” chapter
- “BGP Support for the L2VPN Address Family” chapter

How Enabling VPLS Autodiscovery Differs from Manually Configuring VPLS

With VPLS Autodiscovery enabled, you no longer need to manually set up Virtual Private LAN Service (VPLS). The commands that you use to set up VPLS Autodiscovery are similar to those that you use to manually configure VPLS, as shown in the table below. VPLS Autodiscovery uses **neighbor** commands in L2VPN address family mode to distribute endpoint information to configure a pseudowire.

Table 27: Manual VPLS Configuration Versus VPLS Autodiscovery Configuration

Manual Configuration of VPLS	VPLS Autodiscovery BGP Based

`l2 vfi vpls1 manual`
` vpn id 100`
` neighbor 10.10.10.1 encapsulation mpls`
` neighbor 10.10.10.0 encapsulation mpls`
` exit`
	`l2 vfi vpls1 autodiscovery`
	` vpn id 100`
	` exit`
	`router bgp 1`
	` no bgp default ipv4-unicast`
	` bgp log-neighbor-changes`
	` bgp update-delay 1`
	` neighbor 10.1.1.2 remote-as 1`
	` neighbor 10.1.1.2 update-source Loopback1`
	` .`
	` .`
	` .`
	` address-family l2vpn vpls`
	` neighbor 10.1.1.2 activate`
	` neighbor 10.1.1.2 send-community extended`
	`exit-address-family`

Configure VPLS Autodiscovery by using the **l2 vfi autodiscovery** command. This command allows a virtual forwarding instance (VFI) to learn and advertise pseudowire endpoints. As a result, you no longer need to enter the **neighbor** command in L2 VFI configuration mode.

However, the **neighbor** command is still supported with VPLS Autodiscovery in L2 VFI configuration mode. You can use the **neighbor** command to allow PE devices that do not participate in the autodiscovery process to join the VPLS domain. You can also use the **neighbor** command with PE devices that have been configured using the Tunnel Selection feature. In addition, you can use the **neighbor** command in hierarchical VPLS configurations that have user-facing PE (U-PE) devices that do not participate in the autodiscovery process and have split-horizon forwarding disabled.

show Commands Affected by VPLS Autodiscovery BGP Based

The following **show** commands were enhanced for VPLS Autodiscovery:

- The **show mpls l2transport vc detail** command was updated to include Forwarding Equivalence Class (FEC) 129 signaling information for autodiscovered Virtual Private LAN Service (VPLS) pseudowires.
- The **show vfi** command was enhanced to display information related to autodiscovered virtual forwarding instances (VFIs). The new output includes the VPLS ID, the route distinguisher (RD), the route target (RT), and router IDs of discovered peers.
- The **show xconnect** command was updated with the **rib** keyword to provide Routing Information Base (RIB) information about pseudowires.

BGP VPLS Autodiscovery Support on a Route Reflector

By default, routes received from an internal BGP (iBGP) peer are not sent to another iBGP peer unless a full mesh configuration is formed between all BGP devices within an autonomous system (AS). This results in scalability issues. Using Border Gateway Protocol (BGP) route reflectors leads to much higher levels of scalability. Configuring a route reflector allows a device to advertise or reflect the iBGP learned routes to other iBGP speakers.

Virtual Private LAN Service (VPLS) Autodiscovery supports BGP route reflectors. A BGP route reflector can be used to reflect BGP VPLS prefixes without VPLS being explicitly configured on the route reflector.

A route reflector does not participate in autodiscovery; that is, no pseudowires are set up between the route reflector and the PE devices. A route reflector reflects VPLS prefixes to other PE devices so that these PE devices do not need to have a full mesh of BGP sessions. The network administrator configures only the BGP VPLS address family on a route reflector. For an example configuration of VPLS Autodiscovery support on a route reflector, see the "Example: BGP VPLS Autodiscovery Support on Route Reflector" section.

How to Configure VPLS Autodiscovery BGP Based

Enabling VPLS Autodiscovery BGP Based

Perform this task to enable Virtual Private LAN Service (VPLS) PE devices to discover other PE devices that are part of the same VPLS domain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *vfi-name* **autodiscovery**
4. **vpn id** *vpn-id*
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	**l2 vfi** *vfi-name* **autodiscovery** **Example:** `Device(config)# l2 vfi vpls1 autodiscovery`	Enables VPLS Autodiscovery on a PE device and enters L2 VFI configuration mode.
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 10`	Configures a VPN ID for the VPLS domain.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-vfi)# end`	Exits L2 VFI configuration mode and returns to privileged EXEC mode. • Commands take effect after the device exits L2 VFI configuration mode.

Configuring BGP to Enable VPLS Autodiscovery

The Border Gateway Protocol (BGP) Layer 2 VPN (L2VPN) address family supports a separate L2VPN Routing Information Base (RIB) that contains endpoint provisioning information for Virtual Private LAN Service (VPLS) Autodiscovery. BGP learns the endpoint provisioning information from the L2VPN database, which is updated each time a Layer 2 virtual forwarding instance (VFI) is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **no bgp default ipv4-unicast**
5. **bgp log-neighbor-changes**
6. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
7. **neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. Repeat Steps 6 and 7 to configure other BGP neighbors.
9. **address-family l2vpn** [**vpls**]
10. **neighbor** {*ip-address* | *peer-group-name*} **activate**
11. **neighbor** {*ip-address* | *peer-group-name*} **send-community** {**both** | **standard** | **extended**}
12. Repeat Steps 10 and 11 to activate other BGP neighbors under an L2VPN address family.
13. **exit-address-family**
14. **end**
15. **show vfi**
16. **show ip bgp l2vpn vpls** {**all** | **rd** *route-distinguisher*}

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-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. **Note** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured using the **neighbor remote-as** router configuration command unless you configure the **no bgp default ipv4-unicast** router configuration command before configuring the **neighbor remote-as** command. Existing neighbor configurations are not affected.
Step 5	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 10.10.10.1 remote-as 65000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device. • If the *autonomous-system-number* argument matches the autonomous system number specified in the **router bgp** command, the neighbor is an internal neighbor. • If the *autonomous-system-number* argument does not match the autonomous system number specified in the **router bgp** command, the neighbor is an external neighbor. • In this example, the neighbor at 10.10.10.1 is an internal BGP neighbor.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.10.10.1 update-source loopback1`	(Optional) Configures a device to select a specific source or interface to receive routing table updates. • This example uses a loopback interface. The advantage of this configuration is that the loopback interface is not affected by the effects of a flapping interface.

	Command or Action	Purpose
Step 8	Repeat Steps 6 and 7 to configure other BGP neighbors.	—
Step 9	**address-family l2vpn** [**vpls**] **Example:** `Device(config-router)# address-family l2vpn vpls`	Specifies the L2VPN address family and enters address family configuration mode. • The optional **vpls** keyword specifies that the VPLS endpoint provisioning information is to be distributed to BGP peers. • In this example, an L2VPN VPLS address family session is created.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 10.10.10.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 11	**neighbor** {*ip-address* | *peer-group-name*} **send-community** {**both** | **standard** | **extended**} **Example:** `Device(config-router-af)# neighbor 10.10.10.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • In this example, an extended communities attribute is sent to the neighbor at 10.10.10.1.
Step 12	Repeat Steps 10 and 11 to activate other BGP neighbors under an L2VPN address family.	—
Step 13	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 14	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 15	**show vfi** **Example:** `Device# show vfi`	Displays information about the configured VFI instances.
Step 16	**show ip bgp l2vpn vpls** {**all** | **rd** *route-distinguisher*} **Example:** `Device# show ip bgp l2vpn vpls all`	Displays information about the L2VPN VPLS address family.

Customizing the VPLS Autodiscovery Settings

Several commands allow you to customize the Virtual Private LAN Service (VPLS) environment. You can specify identifiers for the VPLS domain, the route distinguisher (RD), the route target (RT), and the provider edge (PE) device. Perform this task to customize these identifiers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *vfi-name* **autodiscovery**
4. **vpn id** *vpn-id*
5. **vpls-id** {*autonomous-system-number*:*nn* | *ip-address*:*nn*}
6. **rd** {*autonomous-system-number*:*nn* | *ip-address*:*nn*}
7. **route-target** [**import** | **export** | **both**] {*autonomous-system-number*:*nn* | *ip-address*:*nn*}
8. **auto-route-target**
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	**l2 vfi** *vfi-name* **autodiscovery** **Example:** `Device(config)# l2 vfi vpls1 autodiscovery`	Enables VPLS Autodiscovery on the PE device and enters Layer 2 VFI configuration mode.
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 10`	Configures a VPN ID for the VPLS domain.
Step 5	**vpls-id** {*autonomous-system-number*:*nn* | *ip-address*:*nn*} **Example:** `Device(config-vfi)# vpls-id 5:300`	(Optional) Assigns an identifier to the VPLS domain. • This command is optional because VPLS Autodiscovery automatically generates a VPLS ID using the Border Gateway Protocol (BGP) autonomous system (AS) number and the configured VFI VPN ID. You can use this command to change the automatically generated VPLS ID.

	Command or Action	Purpose
		• There are two formats for configuring the VPLS ID argument. It can be configured in the *autonomous-system-number:network number* (*ASN:nn*) format, as shown in the example, or it can be configured in the *IP-address:network number* format (*IP-address:nn*).
Step 6	**rd** {*autonomous-system-number*:*nn* | *ip-address*:*nn*} **Example:** `Device(config-vfi)# rd 2:3`	(Optional) Specifies the RD to distribute endpoint information. • This command is optional because VPLS Autodiscovery automatically generates an RD using the BGP autonomous system number and the configured VFI VPN ID. You can use this command to change the automatically generated RD. • There are two formats for configuring the route distinguisher argument. It can be configured in the *autonomous-system-number:network number* (*ASN:nn*) format, as shown in the example, or it can be configured in the *IP-address:network number* format (*IP-address:nn*).
Step 7	**route-target** [**import** | **export** | **both**] {*autonomous-system-number*:*nn* | *ip-address*:*nn*} **Example:** `Device(config-vfi)# route-target 600:2222`	(Optional) Specifies the RT. • This command is optional because VPLS Autodiscovery automatically generates an RT using the lower 6 bytes of the RD and the VPLS ID. You can use this command to change the automatically generated RT. • There are two formats for configuring the route target argument. It can be configured in the *autonomous-system-number:network number* (*ASN:nn*) format, as shown in the example, or it can be configured in the *IP-address:network number* format (*IP-address:nn*).
Step 8	**auto-route-target** **Example:** `Device(config-vfi)# auto-route-target`	(Optional) Enables the automatic generation of a RT.
Step 9	**end** **Example:** `Device(config-vfi)# end`	Exits L2 VFI configuration mode and returns to privileged EXEC mode. • Commands take effect after the device exits Layer 2 VFI configuration mode.

Configuration Examples for VPLS Autodiscovery BGP Based

The following examples show the configuration of a network that uses VPLS Autodiscovery:

Example: Configuring BGP to Enable VPLS Autodiscovery

PE1

```
l2 router-id 10.1.1.1
l2 vfi auto autodiscovery
 vpn id 100
!
pseudowire-class mpls
 encapsulation mpls
!
interface Loopback1
 ip address 10.1.1.1 255.255.255.255
!
interface GigabitEthernet 0/0/1
 description Backbone interface
 ip address 192.168.0.1 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.0 0.0.0.255 area 0
 network 172.16.0.0 0.0.0.255 area 0
!
router bgp 1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.1.1.2 remote-as 1
 neighbor 10.1.1.2 update-source Loopback1
 neighbor 10.1.1.3 remote-as 1
 neighbor 10.1.1.3 update-source Loopback1
!
 address-family ipv4
 no synchronization
 no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
 neighbor 10.1.1.2 activate
 neighbor 10.1.1.2 send-community extended
 neighbor 10.1.1.3 activate
 neighbor 10.1.1.3 send-community extended
 exit-address-family
```

PE2

```
l2 router-id 10.1.1.2
l2 vfi auto autodiscovery
 vpn id 100
!
 pseudowire-class mpls
 encapsulation mpls
!
interface Loopback1
 ip address 10.1.1.2 255.255.255.255
!
interface GigabitEthernet 0/0/1
 description Backbone interface
 ip address 192.168.0.2 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.0 0.0.0.255 area 0
 network 172.16.0.0 0.0.0.255 area 0
!
router bgp 1
```

```
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 update-source Loopback1
 neighbor 10.1.1.3 remote-as 1
 neighbor 10.1.1.3 update-source Loopback1
!
 address-family ipv4
 no synchronization
 no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.1 send-community extended
 neighbor 10.1.1.3 activate
 neighbor 10.1.1.3 send-community extended
 exit-address-family
```

PE3

```
l2 router-id 10.1.1.3
l2 vfi auto autodiscovery
 vpn id 100
!
pseudowire-class mpls
 encapsulation mpls
!
interface Loopback1
 ip address 10.1.1.3 255.255.255.255
!
interface GigabitEthernet 0/0/1
 description Backbone interface
 ip address 192.168.0.3 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.0 0.0.0.255 area 0
 network 172.16.0.0 0.0.0.255 area 0
!
router bgp 1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 update-source Loopback1
 neighbor 10.1.1.2 remote-as 1
 neighbor 10.1.1.2 update-source Loopback1
!
 address-family ipv4
 no synchronization
 no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.1 send-community extended
 neighbor 10.1.1.2 activate
 neighbor 10.1.1.2 send-community extended
 exit-address-family
```

Example: BGP VPLS Autodiscovery Support on Route Reflector

In the following example, a host named PE-RR (indicating Provider Edge-Route Reflector) is configured as a route reflector that is capable of reflecting Virtual Private LAN Service (VPLS) prefixes. The VPLS address family is configured using the **address-family l2vpn vpls** command.

```
hostname PE-RR
!
router bgp 1
 bgp router-id 10.1.1.3
 no bgp default route-target filter
 bgp log-neighbor-changes
neighbor iBGP-PEERS peer-group
neighbor iBGP-PEERS remote-as 1
neighbor iBGP-PEERS update-source Loopback1
neighbor 10.1.1.1 peer-group iBGP-PEERS
neighbor 10.1.1.2 peer-group iBGP-PEERS
!
address-family l2vpn vpls
  neighbor iBGP-PEERS send-community extended
  neighbor iBGP-PEERS route-reflector-client
  neighbor 10.1.1.1 peer-group iBGP-PEERS
  neighbor 10.1.1.2 peer-group iBGP-PEERS
exit-address-family
```

Additional References

Related Documents

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

Standards and RFCs

Standard and RFC	Title
draft-ietf-l2vpn-signaling-08.txt	*Provisioning, Autodiscovery, and Signaling in L2VPNs*
draft-ietf-l2vpn-vpls-bgp-08.8	*Virtual Private LAN Service (VPLS) Using BGP for Autodiscovery and Signaling*
draft-ietf-mpls-lsp-ping-03.txt	*Detecting MPLS Data Plane Failures*
draft-ietf-pwe3-vccv-01.txt	*Pseudo-Wire (PW) Virtual Circuit Connection Verification (VCCV)*
RFC 3916	*Requirements for Pseudo-wire Emulation Edge-to-Edge (PWE3)*

Standard and RFC	Title
RFC 3981	*Pseudo Wire Emulation Edge-to-Edge Architecture*
RFC 6074	Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)
RFC 4761	Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling

MIBs

MIB	MIBs Link
• CISCO-IETF-PW-ATM-MIB (PW-ATM-MIB) • CISCO-IETF-PW-ENET-MIB (PW-ENET-MIB) • CISCO-IETF-PW-FR-MIB (PW-FR-MIB) • CISCO-IETF-PW-MIB (PW-MIB) • CISCO-IETF-PW-MPLS-MIB (PW-MPLS-MIB)	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

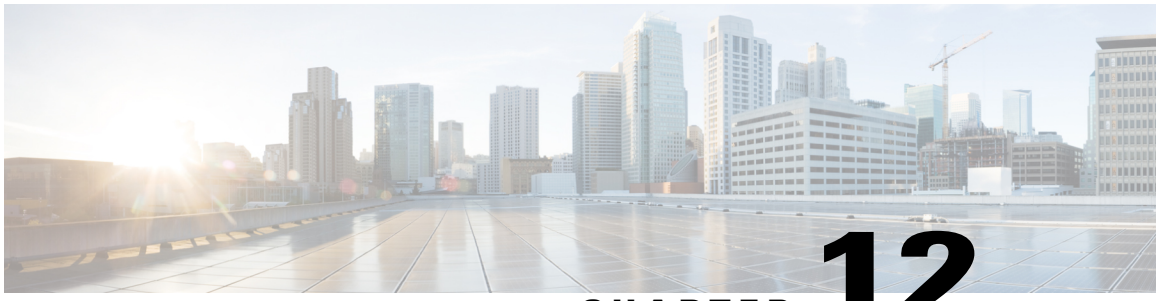
Feature Information for VPLS Autodiscovery BGP Based

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 28: Feature Information for VPLS Autodiscovery BGP Based

Feature Name	Releases	Feature Information
VPLS Autodiscovery BGP Based	Cisco IOS XE Release 3.7S Cisco IOS Release 15.1(1)SY	VPLS Autodiscovery enables each Virtual Private LAN Service (VPLS) provider edge (PE) device to discover other PE devices that are part of the same VPLS domain.



CHAPTER 12

VPLS over GRE

Virtual Private LAN Service (VPLS) enables geographically separate LAN segments to be interconnected as a single bridged domain over an MPLS network (VPLS can only be enabled on an MPLS network).

Finding Feature Information

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

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

Restrictions for VPLS over GRE

- Load-balancing at the VPLS ingress or at the core is not supported for flood or multicast traffic.
- Interior Gateway Protocol (IGP) load balance and flow aware transport of MPLS pseudowires (FAT PW) are not supported for EoMPLS.
- Virtual circuit connection verification (VCCV) over FAT PW is not supported, neither will IGP load balance work for VCCV.

- Configuring scheme 2 of VPLS over GRE by using the **platform vpls gre favor-performance** command is not supported for VPLS/EoMPLS over GRE on MPLS cloud. MPLS should not be enabled on the underlying physical interface that carries the GRE traffic.

Information About VPLS over GRE

VPLS over GRE Overview

Virtual Private LAN Service (VPLS) enables geographically separate LAN segments to be interconnected as a single bridged domain over an MPLS network (VPLS can only be enabled on an MPLS network). Generic routing encapsulation (GRE) is a tunneling protocol developed by Cisco that can encapsulate a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link to Cisco routers at remote points over an IP internetwork. VPLS over GRE then enables VPLS across an IP network. The provider edge (PE) routers for VPLS over GRE must support VPLS and additional GRE encapsulation/decapsulation. The PE routers can be placed in customer sites. For example, different sites of a data center (DC) can have L2 service across an IP network. The PE router can also be placed at the edge of an IP core cloud if a service provider wants to provide L2 service for customers.

A VPLS instance must be configured on each PE router. GRE tunnels are configured to connect PEs across an IP network. MPLS virtual circuit (VC) labels are transported by the MPLS-enabled GRE tunnels. To support the fully meshed pseudowires (PW), GRE tunnels must be fully meshed among PE routers. A pseudowire defines a VLAN and its corresponding pseudoport.

VPLS over GRE Data Plane

In the data plane, the L2 Ethernet frames arrive at the ingress interface on the PE1 router. A VC label is imposed on the Ethernet frame and then the GRE header is encapsulated. An explicit null label could be imposed if the **mpls ldp explicit-null** command is issued on this router. The PE1 router switches the packets to the appropriate interface, which will route the packets to the egress PE2/PE3 routers. When packets arrive at the egress PE2/PE3 routers, the PE2/PE3 routers must decapsulate the GRE header of the IP packets, perform label disposition, L2 lookup, and forward the frame to the appropriate egress interface.

VPLS over GRE Encapsulation

VPLS over GRE requires at least one recirculation at the ingress router (Because of a hardware limitation, hardware cannot encapsulate the VC label + MPLS label + GRE header + L2 rewrite in one packet pass. Packets may travel back to the data path to finish the encapsulation). Packet recirculation is a specific means for packets to travel back to the data path. Two schemes to achieve these recirculations exist.

In scheme one, for remote unicast, two recirculations are required. The first pass handles VC label and MPLS label encapsulation. The hardware must do the recirculation with a shim header indicating the destination index of the GRE tunnel encapsulation adjacency entry. An MTU check is performed in the first pass. The second pass handles GRE encapsulation. In this pass, the GRE header and IP header are added. In addition, the egress features on the GRE tunnel, such as ACL and QoS, are handled in this pass. The hardware must do a second recirculation with a shim header indicating the destination index of the L2 rewrite adjacency entry. The third pass handles L2 rewrite. In the third pass, IPv4 lookup is performed and hits an adjacency that programs a new L2 MAC address.

In scheme two, one recirculation is required. The first pass handles VC label and MPLS label encapsulation. The hardware must do the recirculation. The second pass performs IP + GRE encapsulation and provides a new destination media access control (DMAC). The egress logical interface (LIF) is the physical outgoing interface LIF.

The advantage of scheme two is that scheme two has better performance because of one less pass than scheme one in EARL. The disadvantage of scheme two is that the GRE egress QoS and ACL features are sacrificed.

Scheme one is the default setting in a Cat6k switch. A command is provided to globally change the default setting to scheme two if you want to have better performance. If you select to use scheme two, scheme two only applies to the VC created after the command is issued. If you want to have consistent hardware programming, existing VCs must be brought down and then brought back up.

VPLS over GRE Decapsulation

When packets arrive at the egress router, two recirculations are required. In the first pass, the GRE decapsulation is performed by the layer 3 (L3) module. After the GRE header is removed, the second pass performs EoMPLS decapsulation and the third pass performs L2 lookup and sends out the Ethernet frame to a proper outgoing interface.

VPLS over GRE MTU Requirements

In VPLS over GRE, the PEs are virtually connected by a GRE tunnel. At least one label (4 bytes) and a control word (4 bytes, optional) are added to each frame that is transported across the network. The transport frame is the Ethernet frame, the added 14 bytes are 6 bytes for each source and destination MAC address and 2 bytes for the Ethertype. Finally, 24 bytes are added for the GRE header and the outer IP header.

RFC preferences are to set the tunnel interface descriptor block (IDB) maximum transmission unit (MTU) to be the minimum MTU of all the egress interfaces that can be used by this tunnel to the remote tunnel endpoint. At the ingress router, the MTU size for the first pass should be at least 42 bytes less than the minimum MTU size (12 for MAC destination address [DA] and source address [SA], 2 for Ethertype, 4 for MPLS VC label stack, and 24 for GRE tunnel). 4 bytes for a control word and 4 bytes for an explicit null could be added for certain pseudowires.

EoMPLS over GRE

EoMPLS over GRE is conceptually the same as VPLS over GRE, but it is a peer-to-peer (P2P) service. The first pass decapsulates the GRE header, and the second pass performs EoMPLS decapsulation and sends the traffic to the proper interface.

How to Configure VPLS over GRE

Configuring VPLS over GRE

Perform these steps to configure VPLS over GRE on your Cisco network. If you would like to enable scheme two of VPLS over GRE, use the **platform vpls gre favor-performance** command at the end of these steps.

Note In scenarios where Generic Routing Encapsulation (GRE) is implemented over multiple Equal-Cost Multipath (ECMP) routes, and scheme two of VPLS over GRE is configured by using the **platform vpls gre favor-performance** command, the following should be considered. Scheme two of VPLS over GRE selects one of the ECMP routes as egress. Additional logic is executed on the supervisor engine or line card while selecting an ECMP route. Each supervisor engine or line card can select different ECMP routes as egress. For example, if GRE has two possible ECMP routes, the supervisor engine may select one route while the line card may select the other route as egress.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type/slot/port*
4. **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]]
5. **exit**
6. **interface** *type/slot/port*
7. **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]]
8. **mpls ip**
9. **tunnel source** {*ip-address* | *type/number*}
10. **tunnel destination** {*hostname* | *ip-address*}
11. **exit**
12. **interface** *type/slot/port*
13. **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]]
14. **mpls ip**
15. **tunnel source** {*ip-address* | *type/number*}
16. **tunnel destination** {*hostname* | *ip-address* }
17. **exit**
18. **ip route** [**vrf** *vrf-name*] *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**dhcp**] [*distance*] [**name** *next-hop-name*] [**permanent** |**track** *number*] [**tag** *tag*]
19. **ip route** [**vrf** *vrf-name*] *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**dhcp**] [*distance*] [**name** *next-hop-name*] [**permanent** | **track** *number*] [**tag** *tag*]
20. **l2 vfi** *name* **manual**
21. **vpn id** *vpn-id*
22. **neighbor** *ip-address* [*vc-id*] {**encapsulation mpls** | **pw-class** *pw-class-name* } [*no-split-horizon*]
23. **neighbor** *ip-address* [*vc-id*] { **encapsulation mpls** | **pw-class** *pw-class-name* } [*no-split-horizon*]
24. **exit**
25. **interface** *type number*
26. **switchport mode access**
27. **switchport access vlan** *vlan-id*
28. **interface vlan** *vlan-id*
29. **xconnect vfi** *vfi-name*
30. **exit**
31. **platform vpls gre favor-performance**

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type/slot/port* **Example:** `Device(config)# interface Loopback0`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]] **Example:** `Device(config-if)# ip address 209.165.202.225 255.255.255.224`	Sets a primary or secondary IP address for an interface.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 6	**interface** *type/slot/port* **Example:** `Device(config)# interface Tunnel0`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]] **Example:** `Device(config-if)# ip address 209.165.200.225 255.255.255.224`	Sets a primary or secondary IP address for an interface.
Step 8	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables Multiprotocol Label Switching (MPLS) forwarding of IPv4 packets along normally routed paths for a particular interface.
Step 9	**tunnel source** {*ip-address* | *type/number*} **Example:** `Device(config-if)# tunnel source 209.165.201.1`	Configures the tunnel source.

	Command or Action	Purpose
Step 10	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Device(config-if)# tunnel destination 209.165.201.2`	Configures the tunnel destination.
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 12	**interface** *type/slot/port* **Example:** `Device(config)# interface Tunnel1`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 13	**ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]] **Example:** `Device(config-if)# ip address 209.165.201.3 255.255.255.224`	Sets a primary or secondary IP address for an interface.
Step 14	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables Multiprotocol Label Switching (MPLS) forwarding of IPv4 packets along normally routed paths for a particular interface.
Step 15	**tunnel source** {*ip-address* | *type/number*} **Example:** `Device(config-if)# tunnel source 209.165.201.4`	Configures the tunnel source.
Step 16	**tunnel destination** {*hostname* | *ip-address* } **Example:** `Device(config-if)# tunnel destination 209.165.201.5`	Configures the tunnel destination.
Step 17	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 18	**ip route** [**vrf** *vrf-name*] *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**dhcp**] [*distance*] [**name** *next-hop-name*] [**permanent** |**track** *number*] [**tag** *tag*] **Example:** `Device(config)# ip route 209.165.201.6 255.255.255.224 Tunnel0`	Establishes a static route.

	Command or Action	Purpose
Step 19	**ip route** [**vrf** *vrf-name*] *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**dhcp**] [*distance*] [**name** *next-hop-name*] [**permanent** | **track** *number*] [**tag** *tag*] **Example:** `Device(config)# ip route 209.165.201.7 255.255.255.255 Tunnel1`	Establishes another static route.
Step 20	**l2 vfi** *name* **manual** **Example:** `Device(config)# l2 vfi green manual`	Creates a Layer 2 virtual forwarding instance (VFI) and enters Layer 2 manual configuration mode.
Step 21	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 100`	Configures a VPN ID for a VPLS domain. The emulated VCs bound to this Layer 2 VRF use this VPM ID for signaling.
Step 22	**neighbor** *ip-address* [*vc-id*] {**encapsulation mpls** | **pw-class** *pw-class-name* } [*no-split-horizon*] **Example:** `Device(config-vfi)# neighbor 209.165.201.7 encapsulation mpls`	Specifies the router that should form a point-to-point Layer 2 virtual forwarding interface (VFI) connection.
Step 23	**neighbor** *ip-address* [*vc-id*] { **encapsulation mpls** | **pw-class** *pw-class-name* } [*no-split-horizon*] **Example:** `Device(config-vfi)# neighbor 209.165.201.6 encapsulation mpls`	Specifies the router that should form a point-to-point Layer 2 virtual forwarding interface (VFI) connection.
Step 24	**exit** **Example:** `Device(config-if)# exit`	Exits Layer 2 manual configuration mode.
Step 25	**interface** *type number* **Example:** `Device(config)# interface gigabitEthernet 5/23`	Selects an interface to configure and enters interface configuration mode.
Step 26	**switchport mode access** **Example:** `Device(config-if)# switchport mode access`	Sets the interface type to nontrunking, nontagged single VLAN Layer 2 interface.

	Command or Action	Purpose
Step 27	**switchport access vlan** *vlan-id* **Example:** `Device(config-if)# switchport access vlan 100`	Sets the VLAN when the interface is in access mode and enters Layer 2 manual configuration mode.
Step 28	**interface vlan** *vlan-id* **Example:** `Device(config-vfi)# interface vlan 100`	Creates or accesses a dynamic switched virtual interface (SVI).
Step 29	**xconnect vfi** *vfi-name* **Example:** `Device(config-vfi)# xconnect vfi green`	Specifies a Layer 2 VFI that you are binding to the VLAN port.
Step 30	**exit** **Example:** `Device(config-vfi)# exit`	Exits Layer 2 manual configuration mode and returns to global configuration mode.
Step 31	**platform vpls gre favor-performance** **Example:** `Device(config)# platform vpls gre favor-performance`	Optional step to enable scheme 2 of VPLS over GRE.

Configuration Examples for VPLS over GRE

Example: Configuring VPLS over GRE

The following example enables scheme one of VPLS over GRE, which is the default. To enable scheme two, use the **platform vpls gre favor-performance** command after all these commands.

PE1

```
Device(config)# interface Loopback0
Device(config-if)# ip address 209.165.202.225 255.255.255.224

Device(config)# interface Tunnel0
Device(config-if)# ip address 209.165.200.225 255.255.255.224
Device(config-if)# mpls ip
Device(config-if)# tunnel source 209.165.201.1
Device(config-if)# tunnel destination 209.165.201.2

Device(config)# interface Tunnel1
Device(config-if)# ip address 209.165.201.3 255.255.255.224
Device(config-if)# mpls ip
Device(config-if)# tunnel source 209.165.201.4
```

```
Device(config-if)# tunnel destination 209.165.201.5

Device(config)# ip route 209.165.201.6 255.255.255.224 Tunnel0
Device(config)# ip route 209.165.201.7 255.255.255.224 Tunnel1


Device(config)# l2 vfi green manual
Device(config-vfi)# vpn id 100
Device(config-vfi)# neighbor 209.165.201.7 encapsulation mpls
Device(config-vfi)# neighbor 209.165.201.6 encapsulation mpls


Device(config)# int gigabitEthernet 5/23
Device(config-if)# switchport mode access
Device(config-if)# switchport access vlan 100


Device(config-vfi)# interface Vlan 100
Device(config-if)# xconnect vfi green
```

PE2

```
Device2(config-if)# interface Loopback0
Device2(config-if)# ip address 209.165.201.6 255.255.255.224

Device2(config-if)# interface Tunnel0
Device2(config-if)# ip address 209.165.201.8 255.255.255.224
Device2(config-if)# mpls ip
Device2(config-if)# tunnel source 209.165.201.2
Device2(config-if)# tunnel destination 209.165.201.1

Device2(config-if)# interface Tunnel1
Device2(config-if)# ip address 209.165.201.9 255.255.255.224
Device2(config-if)# mpls ip
Device2(config-if)# tunnel source 209.165.201.10
Device2(config-if)# tunnel destination 209.165.201.11

Device2(config)# ip route 209.165.202.224 255.255.255.224 Tunnel0
Device2(config)# ip route 209.165.201.7 255.255.255.255 Tunnel1

Device2(config)# l2 vfi green manual
Device2(config-vfi)# vpn id 100
Device2(config-vfi)# neighbor 209.165.202.225 encapsulation mpls
Device2(config-vfi)# neighbor 209.165.201.7 encapsulation mpls

Device2(config)# int gigabitEthernet 5/23
Device2(config-if)# switchport mode access
Device2(config-if)# switchport access vlan 100

Device2(config)# interface Vlan 100
Device2(config-if)# xconnect vfi green
```

Additional References for VPLS over GRE

Related Documents

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

Related Topic	Document Title
MPLS commands	Multiprotocol Label Switching Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 4762	*Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*

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 VPLS over GRE

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 29: Feature Information for Configuring Scheme Two of VPLS over GRE

Feature Name	Releases	Feature Information
VPLS over GRE	15.1(1)SY	The VPLS over GRE feature. The following commands were introduced or modified: **platform vpls gre favor-performance**