Implementing MPLS Layer 2 VPNs

This module provides the conceptual and configuration information for MPLS Layer 2 virtual private networks (VPNs) on Cisco IOS XR software.

For the functionality of MPLS VPNs over IP Tunnels, see Implementing MPLS VPNs over IP Tunnels in *Cisco IOS XR Virtual Private Network Configuration Guide*.

Note For more information about MPLS Layer 2 VPN on the Cisco IOS XR software and for descriptions of the commands listed in this module, see the "Related Documents" section. To locate documentation for other commands that might appear while executing a configuration task, search online in the Cisco IOS XR software master command index.

Feature History for Implementing MPLS Layer 2 VPN Configuration Module

Release	Modification
Release 3.4.0	This feature was introduced.
Release 3.4.1	Support was added for: • Virtual Circuit Connection Verification (VCCV) on L2VPN • QinQ mode and QinAny mode for EoMPLS
Release 3.5.0	Support was added for: • EoMPLS Inter-AS mode • Mac-in-Mac protocol
Release 3.6.0	Support was added for: • Ethernet Remote Port Shutdown

Release 3.7.0	Support was added for ATM over MPLS (ATMoMPLS) with Layer 2VPN capability.
Release 3.8.0	Support was added for Any Transport over MPLS (AToM) for the following features:

- IP Interworking on Engine 3 and 5 Line Cards

- PPP/HDLC Like-to-Like Pseudowires on Engine 3 and Engine 5 Line Cards

- ATM Like-to-Like Pseudowires on Engine 3 and Engine 5 Line Cards

- Frame Relay DLCI, and MLFR Like-to-Like Pseudowires on Engine 3 Line Cards

- Ethernet Port Mode and VLAN Like-to-Like Pseudowires on Engine 3 Line Cards

- Local Switching Support with L2TPv3 on Engine 3 and Engine 5 Line Cards

Support was added for QinQ mode and QinAny mode for EoMPLS on the Cisco CRS.

Contents

Prerequisites for Implementing MPLS L2VPN

To perform these configuration tasks, your Cisco IOS XR software system administrator must assign you to a user group associated with a task group that includes the corresponding command task IDs. All command task IDs are listed in individual command references and in the *Cisco IOS XR Task ID Reference Guide*.

If you need assistance with your task group assignment, contact your system administrator.

Information About Implementing L2VPN

To implement MPLS L2VPN, you should understand the following concepts:

L2VPN Overview

Layer 2 VPN (L2VPN) emulates the behavior of a LAN across an IP or MPLS-enabled IP network allowing Ethernet devices to communicate with each other as they would when connected to a common LAN segment.

As Internet service providers (ISPs) look to replace their Frame Relay or Asynchronous Transfer Mode (ATM) infrastructures with an IP infrastructure, there is a need for to provide standard methods of using an IP infrastructure to provide a serviceable L2 interface to customers; specifically, to provide standard ways of using an IP infrastructure to provide virtual circuits between pairs of customer sites.

Building a L2VPN system requires coordination between the ISP and the customer. The ISP provides L2 connectivity; the customer builds a network using data link resources obtained from the ISP. In an L2VPN service, the ISP does not require information about a the customer's network topology, policies, routing information, point-to-point links, or network point-to-point links from other ISPs.

The ISP requires provider edge (PE) routers with the following capabilities:

- Encapsulation of L2 protocol data units (PDU) into Layer 3 (L3) packets.
- Interconnection of any-to-any L2 transports.
- Emulation of L2 quality-of-service (QoS) over a packet switch network.
- Ease of configuration of the L2 service.
- Support for different types of tunneling mechanisms (MPLS, L2TPv3, IPSec, GRE, and others).
- L2VPN process databases include all information related to circuits and their connections.

ATMoMPLS with L2VPN Capability

These topics describe the ATM over MPLS (ATMoMPLS) with L2VPN feature:

ATMoMPLS with L2VPN Overview

The ATMoMPLS feature supports ATM Adaptation Layer 5 (AAL5) transport. ATMoMPLS is a type of Layer 2 point-to-point connection over an MPLS core. ATMoMPLS and ATM local switching are supported only for ATM-to-ATM interface-to-interface switching combinations.

To implement the ATMoMPLS feature, the Cisco CRS-1 router plays the role of provider edge (PE) router at the edge of a provider network in which customer edge (CE) devices are connected to the Cisco CRS-1 routers.

Layer 2 Local Switching Overview

Local switching lets you to switch Layer 2 data between two interfaces of the same type (for example, ATM-to-ATM, or Frame Relay-to-Frame Relay) or between interfaces of different types (for example, Frame Relay to ATM) on the same router, over an IP core network. The interfaces are on the same line card or on two different cards. During these types of switching, Layer 2 address is used instead of the Layer 3 address.

In addition, same-port local switching lets you to switch Layer 2 data between two circuits on the same interface.

ATM Adaptation Layer 5

AAL5 lets you transport AAL5 PDUs from various customers over an MPLS backbone. ATM AAL5 extends the usability of the MPLS backbone by enabling it to offer Layer 2 services in addition to already existing Layer 3 services. You can enable the MPLS backbone network to accept AAL5 PDUs by configuring the provider edge (PE) routers at both ends of the MPLS backbone.

To transport AAL5 PDUs over MPLS, a virtual circuit is set up from the ingress PE router to the egress PE router. This virtual circuit transports the AAL5 PDUs from one PE router to the other. Each AAL5 PDU is transported as a single packet.

Virtual Circuit Connection Verification on L2VPN

Virtual Circuit Connection Verification (VCCV) is an L2VPN Operations, Administration, and Maintenance (OAM) feature that allows network operators to run IP-based provider edge-to-provider edge (PE-to-PE) keepalive protocol across a specified pseudowire to ensure that the pseudowire data path forwarding does not contain any faults. The disposition PE receives VCCV packets on a control channel, which is associated with the specified pseudowire. The control channel type and connectivity verification type, which are used for VCCV, are negotiated when the pseudowire is established between the PEs for each direction.

Two types of packets can arrive at the disposition egress:

- Type 1—Specifies normal Ethernet-over-MPLS (EoMPLS) data packets.
- Type 2—Specifies VCCV packets.

Cisco IOS XR software supports Label Switched Path (LSP) VCCV Type 1, which uses an inband control word if enabled during signaling. The VCCV echo reply is sent as IPv4 that is the reply mode in IPv4. The reply is forwarded as IP, MPLS, or a combination of both.

VCCV pings counters that are counted in MPLS forwarding on the egress side. However, on the ingress side, they are sourced by the route processor and do not count as MPLS forwarding counters.

Ethernet over MPLS

Ethernet-over-MPLS (EoMPLS) provides a tunneling mechanism for Ethernet traffic through an MPLS-enabled L3 core and encapsulates Ethernet protocol data units (PDUs) inside MPLS packets (using label stacking) to forward them across the MPLS network.

EoMPLS features are described in the following subsections:

- Cisco IOS XR VLAN Mode, page VPC-19
- Inter-AS Mode, page VPC-20
- QinQ Mode, page VPC-21
- QinAny Mode, page VPC-22
- Mac-in-Mac Protocol (Provide Backbone Bridging), page VPC-22

Ethernet Port Mode

In Ethernet port mode, both ends of a pseudowire are connected to Ethernet ports. In this mode, the port is tunneled over the pseudowire or, using local switching (also known as an *attachment circuit-to-attachment circuit cross-connect*) switches packets or frames from one attachment circuit (AC) to another AC attached to the same PE node.

> **Note** L2VPN forwarding using GRE tunnels is supported in the Ethernet port mode.

Figure 1 provides an example of Ethernet port mode.

Figure 1   Ethernet Port Mode Packet Flow
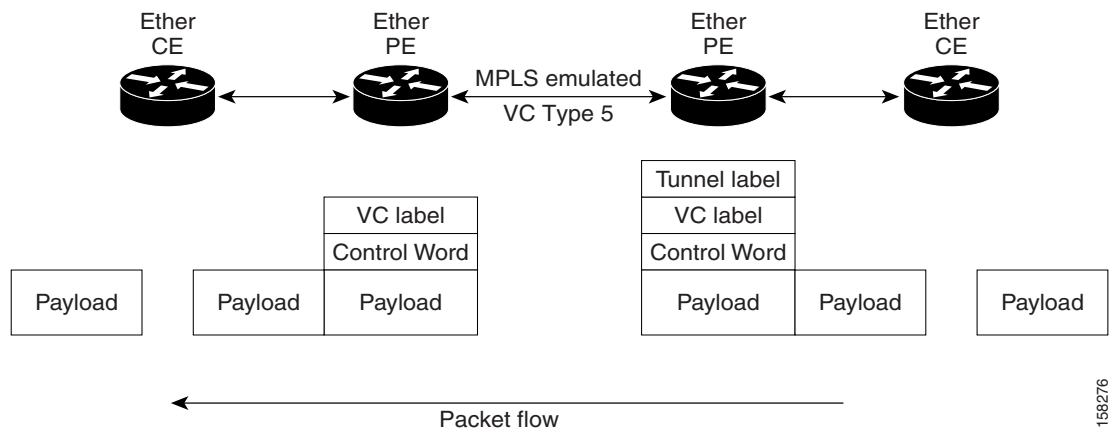
Cisco IOS XR VLAN Mode

In VLAN mode, each VLAN on a customer-end to provider-end link can be configured as a separate L2VPN connection using virtual connection (VC) type 4 or VC type 5. VC type 4 is the default mode.

As illustrated in Figure 2, the Ethernet PE associates an internal VLAN-tag to the Ethernet port for switching the traffic internally from the ingress port to the pseudowire; however, before moving traffic into the pseudowire, it removes the internal VLAN tag.

Figure 2   *VLAN Mode Packet Flow*
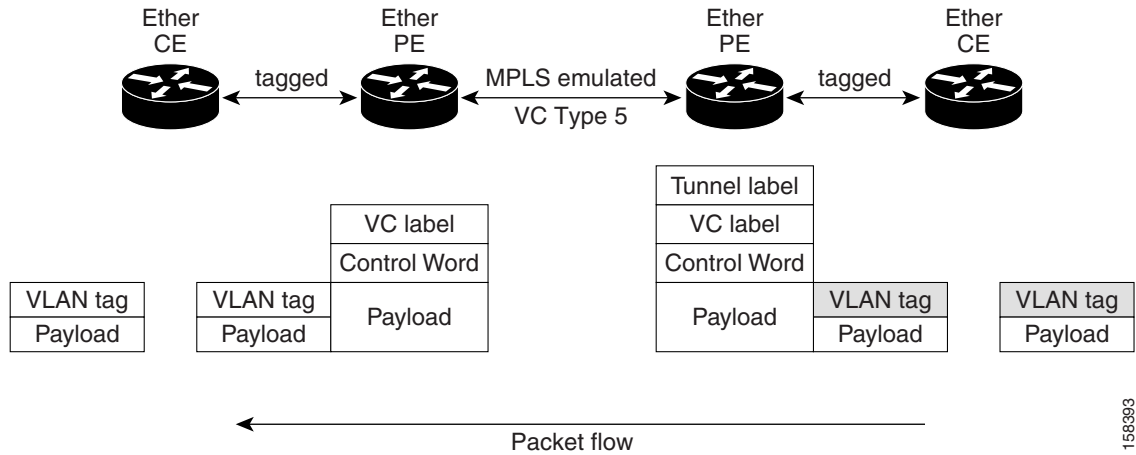
At the egress VLAN PE, the PE associates a VLAN tag to the frames coming off of the pseudowire and after switching the traffic internally, it sends out the traffic on an Ethernet trunk port.

Note   Because the port is in trunk mode, the VLAN PE doesn't remove the VLAN tag and forwards the frames through the port with the added tag.

Note   L2VPN forwarding using GRE tunnels is supported in the VLAN mode.

Inter-AS Mode

Inter-AS is a peer-to-peer type model that allows extension of VPNs through multiple provider or multi-domain networks. This lets service providers peer up with one another to offer end-to-end VPN connectivity over extended geographical locations.

EoMPLS support can assume a single AS topology where the pseudowire connecting the PE routers at the two ends of the point-to-point EoMPLS cross-connects resides in the same autonomous system; or multiple AS topologies in which PE routers can reside on two different ASs using iBGP and eBGP peering.

Figure 3 illustrates MPLS over Inter-AS with a basic double AS topology with iBGP/LDP in each AS.

Figure 3   *EoMPLS over Inter-AS: Basic Double AS Topology*
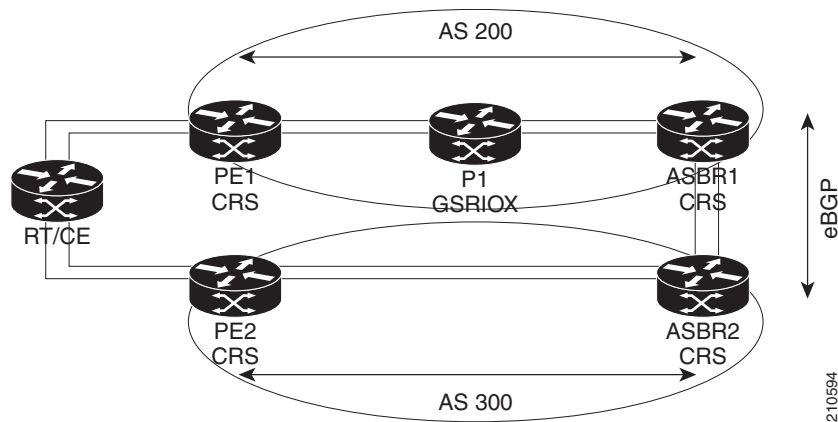
QinQ Mode

QinQ is an extension of 802.1Q for specifying multiple 802.1Q tags (IEEE 802.1QinQ VLAN Tag stacking). Layer 3 VPN service termination and L2VPN service transport are enabled over QinQ sub-interfaces.

The Cisco CRS-1 router implements the Layer 2 tunneling or Layer 3 forwarding depending on the subinterface configuration at provider edge routers. This function only supports up to two QinQ tags on the SPA and fixed PLIM:

- Layer 2 QinQ VLANs in L2VPN attachment circuit: QinQ L2VPN attachment circuits are configured under the Layer 2 transport subinterfaces for point-to-point EoMPLS based cross-connects using both virtual circuit type 4 and type 5 pseudowires and point-to-point local-switching-based cross-connects including full interworking support of QinQ with 802.1q VLANs and port mode.

- Layer 3 QinQ VLANs: Used as a Layer 3 termination point, both VLANs are removed at the ingress provider edge and added back at the remote provider edge as the frame is forwarded.

Layer 3 services over QinQ include:

- IPv4 unicast and multicast

- IPv6 unicast and multicast

- MPLS

- Connectionless Network Service (CLNS) for use by Intermediate System-to-Intermediate System (IS-IS) Protocol

**Note**   The Cisco CRS-1 router does not support: bundle attachment circuits and Hot Standby Router Protocol (HSRP) or Virtual Router Redundancy Protocol (VRRP) on QinQ subinterfaces.

In QinQ mode, each CE VLAN is carried into an SP VLAN. QinQ mode should use VC type 5, but VC type 4 is also supported. On each Ethernet PE, you must configure both the inner (CE VLAN) and outer (SP VLAN).

Figure 4 illustrates QinQ using VC type 4.

Figure 4   *EoMPLS over QinQ Mode*
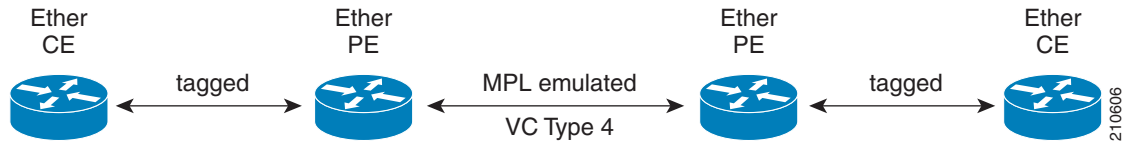
QinAny Mode

In the QinAny mode, the service provider VLAN tag is configured on both the ingress and the egress nodes of the provider edge VLAN. QinAny mode is similar to QinQ mode using a Type 5 VC, except that the customer edge VLAN tag is carried in the packet over the pseudowire, as the customer edge VLAN tag is unknown.

Mac-in-Mac Protocol (Provide Backbone Bridging)

The Mac-in-Mac (or, Provider Backbone Bridging) protocol lets service providers scale networks using Ethernet technology to maintain management and operational simplicity, and reduce operating costs.

Mac-In-Mac encapsulates the customer MAC header with a service provider MAC header. Instead of using additional Q-tags to separate end customers, a 24-bit service tag in the service provider encapsulating MAC header is used, which provides support for up to 16-million service instances.

Note   Mac-In-Mac is standardized as IEEE 802.1ah.

Quality of Service

Using L2VPN technology, you can assign a quality of service (QoS) level to both Port and VLAN modes of operation.

L2VPN technology requires that QoS functionality on PE routers be strictly L2-payload-based on the edge-facing interfaces (also know as *attachment circuits*). Figure 5 illustrates L2 and L3 QoS service policies in a typical L2VPN network.

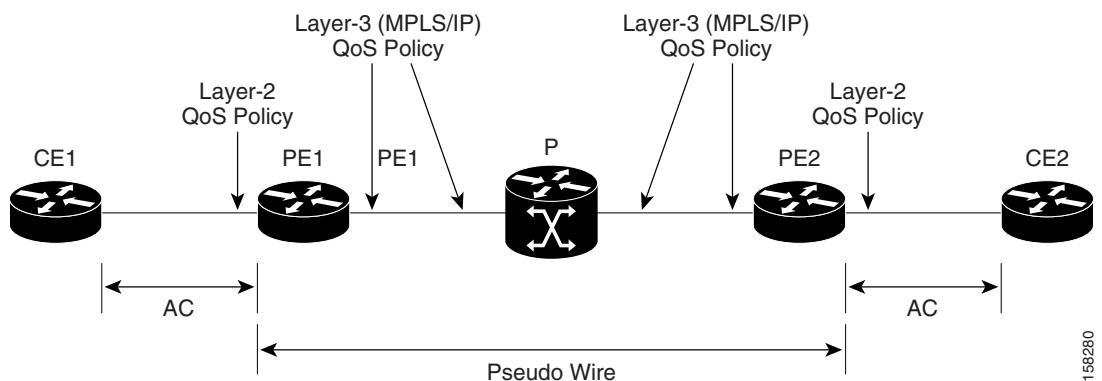*Figure 5*   *L2VPN QoS Feature Application*

Figure 6 shows four packet processing paths within a provider edge device where a QoS service policy can be attached. In an L2VPN network, packets are received and transmitted on the edge-facing interfaces as L2 packets and transported on the core-facing interfaces as MPLS (EoMPLS) or IP (L2TP) packets.

Figure 6   *L2VPN QoS Reference Model*
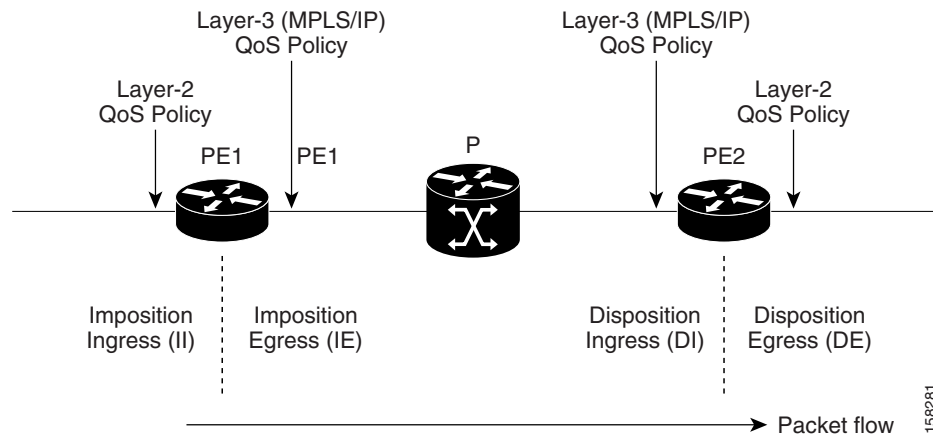
High Availability

L2VPN uses control planes in both route processors and line cards, as well as forwarding plane elements in the line cards.

Note The l2tp_mgr process does not support high availability.

The availability of L2VPN meets the following requirements:

- A control plane failure in either the route processor or the line card will not affect the circuit forwarding path.
- The router processor control plane supports failover without affecting the line card control and forwarding planes.
- L2VPN integrates with existing Label Distribution Protocol (LDP) graceful restart mechanism.

Preferred Tunnel Path

Preferred tunnel path functionality lets you map pseudowires to specific traffic-engineering tunnels. Attachment circuits are cross-connected to specific MPLS traffic engineering tunnel interfaces instead of remote PE router IP addresses (reachable using IGP or LDP). Using preferred tunnel path, it is always assumed that the traffic engineering tunnel that transports the L2 traffic runs between the two PE routers (that is, its head starts at the imposition PE router and its tail terminates on the disposition PE router).

Note
- Currently, preferred tunnel path configuration applies only to MPLS encapsulation.
- The fallback enable option is supported.

Any Transport over MPLS

Any Transport over MPLS (AToM) transports Layer 2 packets over a Multiprotocol Label Switching (MPLS) backbone, which enables service providers to connect customer sites with existing Layer 2 networks by using a single, integrated, packet-based network infrastructure. Using this feature, service providers can deliver Layer 2 connections over an MPLS backbone, instead of using separate networks.

AToM encapsulates Layer 2 frames at the ingress PE router and sends them to a corresponding PE router 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 set up the connection, called a *pseudowire*, between the routers. You 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

These topics describe the AToM feature:

IP Interworking

In AToM IP Interworking, also called *routed interworking*, the carrier edge (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.

IP Interworking 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.

The following modes support IP Interworking on AToM:

- ATM to Ethernet: In this interworking, both ATM and Ethernet PE routers are configured for IP interworking. IP packets from an ATM CE are encapsulated using IP over MPLS and trasmitted over the pseudowire. On the Ethernet side, the Ethernet PE removes the Layer 2 framing on the Ethernet packets from the Ethernet CE and forwards the IP packet on the pseudowire using IP over MPLS encapsulation. Non-IP packets are dropped in this process. At the ATM PE, after label disposition, the IP packets are encapsulated over AAL5 using IP encapsulation. In either direction, packets for which translations are not supported, are dropped.
- Ethernet port to VLAN mode: Using the Ethernet port mode, you can create an Ethernet virtual local area network (VLAN) among geographically separated sites. Different sites can operate together over an MPLS network as though they were on a common Ethernet network.
- Frame Relay to Ethernet: Multi-protocol Frame Relay packets from the Frame Relay CE are encapsulated using IP over MPLS and transmitted over the pseudowire. On the Ethernet side, the Ethernet PE removes the Layer 2 framing on the Ethernet packets from the Ethernet CE and forwards

the Layer 3 packet over the pseudowire using IP over MPLS encapsulation. At the Frame Relay PE, after label disposition, the Layer 3 packets are encapsulated over Frame Relay using IP encapsulation. In either direction, packets for which translations are not supported are dropped.

- Frame Relay to ATM AAL5: ATM and Frame Relay links are locally terminated and IP interworking is used to transport the Layer 3 packets over the IP over MPLS pseudowire.
- ATM AAL5 - ATM Adaptation Layer Type-5 (AAL5) allows efficient transportation of PVCs across the MPLS backbone. Multiple PVCs can be multiplexed onto a single label switched path between the provider edge routers.

The following types of cross connections are supported for AToM IP Interworking:

- Ethernet
  - VLAN
  - Q-in-Q
  - Frame Relay
  - ATM AAL5 SNAP/MUX/NLPID
- VLAN
  - Ethernet
  - Q-in-Q
  - Frame Relay
  - ATM AAL5 SNALP/MUX/NLPID
- Q-in-Q
  - Ethernet
  - VLAN
  - Frame Relay
  - ATM AAL5 SNAP/MUX/NLPID
- Frame Relay
  - Ethernet
  - VLAN
  - Q-in-Q
  - ATM AAL5 SNAP/MUX/NLPID

Control Word Processing

The control word contains forward explicit congestion notification (FECN), backward explicit congestion notification (BECN) and DE bits in case of frame relay connection.

Control word is mandatory for the following:

- Frame Relay
- ATM AAL5

The system does not map bits from one transport end point to another across an AToM IP Interworking connection.

Like-to-Like Pseudowires

A pseudowire (PW) is a bidirectional VC connecting two Attached Circuits. In an MPLS network, PWs are carried inside an LSP tunnel.

A point-to-point (PPP) connection allows service providers to provide a transparent PPP pass-through where the customer-edge routers can exchange the traffic through an end-to-end PPP session. Service providers can offer a virtual leased-line solution, and use the PPP subinterface capability to peer with multiple providers through a single POS connection.

A High-Level Data Link control (HDLC) connection is emulated from a customer router to another customer router across an MPLS backbone. This technology allows transportation of HDLC frames across the packet networks. HDLC over MPLS also works in transparent mode.

How to Implement L2VPN

This section describes the tasks required to implement L2VPN:

- Configuring an Interface or Connection for L2VPN, page VPC-26
- Configuring Static Point-to-Point Cross-Connects, page VPC-29
- Configuring Dynamic Point-to-Point Cross-Connects, page VPC-31
- Configuring Inter-AS, page VPC-33
- Configuring L2VPN Quality of Service, page VPC-33
- Configuring Preferred Tunnel Path, page VPC-39
- Configuring AToM IP Interworking, page VPC-41

Configuring an Interface or Connection for L2VPN

Perform this task to configure an interface or a connection for L2VPN.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **l2transport**
4. **exit**
5. **interface** *type interface-path-id*
6. **dot1q native vlan** *vlan-id*
7. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`RP/0/RP0/CPU0:router# configure`
		Enters global configuration mode.

Step 2	`interface` *type* *interface-path-id*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# interface`
	`GigabitEthernet 0/0/0/0`
		Enters interface configuration mode and configures an interface.

Step 3	`l2transport`
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# l2transport`
		Enables L2 transport on the selected interface.

Step 4	`exit`
	
	Example:
	`RP/0/RP0/CPU0:router(config-if-l2)# exit`
		Exits the current configuration mode.

Step 5	`interface` *type* *interface-path-id*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# interface`
	`GigabitEthernet0/0/0/0`
		Enters interface configuration mode and configures an interface.

	Command or Action	Purpose
Step 6	**dot1q native vlan** *vlan ID* **Example:** RP/0/RP0/CPU0:router(config-if)# dot1q vlan 1	Assigns the native VLAN ID of a physical interface trunking 802.1Q VLAN traffic.

Step 7	**end**
	or
	commit
	
	Example:
	RP/0/RP0/CPU0:router(config-if)# end
	or
	RP/0/RP0/CPU0:router(config-if)# commit
		Saves configuration changes.
		
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before`
		`exiting(yes/no/cancel)?`
		`[cancel]:`
		
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Static Point-to-Point Cross-Connects

Perform this task to configure static point-to-point cross-connects.

Please consider the following information about cross-connects when you configure static point-to-point cross-connects:

- An cross-connect is uniquely identified with the pair; the cross-connect name must be unique within a group.
- A segment (an attachment circuit or pseudowire) is unique and can belong only to a single cross-connect.
- A static VC local label is globally unique and can be used in one pseudowire only.
- No more than 16,000 cross-connects can be configured per router.

Note Static pseudowire connections do not use LDP for signaling.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*
4. **p2p** *xconnect-name*
5. **interface** *type interface-path-id*
6. **neighbor** *ip-address* **pw-id** *pseudowire-id*
7. **mpls static label local** {*value*} **remote** {*value*}
8. **end**
   or
   commit
9. **show l2vpn xconnect group** *group name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**l2vpn** **Example:** `RP/0/RP0/CPU0:router(config)# l2vpn`	Enters L2VPN configuration mode.

	Command or Action	Purpose
Step 3	**xconnect group** *group name* **Example:** RP/0/RP0/CPU0:router(config-l2vpn)# xconnect group vlan_grp_1	Enters the name of the cross-connect group.
Step 4	**p2p** *xconnect name* **Example:** RP/0/RP0/CPU0:router(config-l2vpn-xc)# p2p vlan1	Enters a name for the point-to-point cross-connect.
Step 5	**interface** *type interface-path-id* **Example:** RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)# interface GigabitEthernet0/0/0/0.1	Specifies the interface type ID. The choices are: • GigabitEthernet: GigabitEthernet/IEEE 802.3 interfaces. • TenGigE: TenGigabitEthernet/IEEE 802.3 interfaces.
Step 6	**neighbor** *ip-address* **pw-id** *pseudowire-id* **Example:** RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)# neighbor 2.2.2.2 pw-id 2000	Configures the pseudowire segment for the cross-connect. Optionally, you can disable the control word or set the transport-type to "Ethernet" or "VLAN".
Step 7	**mpls static label local** {*value*} **remote** {*value*} **Example:** RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# mpls static label local 699 remote 890	Configures local and remote label ID values.

	Command or Action	Purpose
Step 8	**end** or **commit** **Example:** RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# e nd or RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# commit	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 9	**show l2vpn xconnect group** *group name* **Example:** RP/0/RP0/CPU0:show l2vpn xconnect group p2p	Displays the name of the Point-to-Point cross-connect group you created.

Configuring Dynamic Point-to-Point Cross-Connects

Perform this task to configure dynamic point-to-point cross-connects.

Note For dynamic cross-connects, LDP must be up and running.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*
4. **p2p** *xconnect-name*
5. **interworking ipv4**
6. **interface** *type interface-path-id*
7. **neighbor** *ip-address* **pw-id** *pseudowire-id*
8. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`RP/0/RP0/CPU0:router# configure`
		Enters the configuration mode.

Step 2	`l2vpn`
	
	Example:
	`RP/0/RP0/CPU0:router(config)# l2vpn`
		Enters L2VPN configuration mode.

Step 3	`xconnect group` *group-name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn)# xconnect group grp_1`	Enters the name of the cross-connect group.
Step 4	`p2p` *xconnect-name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-xc)# p2p vlan1`	Enters a name for the point-to-point cross-connect.

Step 5	`interworking ipv4`
	
	Example:
	`RP/0/RP0/CPU0:router(config-l2vpn-xc)# interworking ipv4`
		Configure the interworking for IPv4 network.

Step 6	`interface` *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)# interface GigabitEthernet0/0/0/0.1`	Specifies the interface type ID. The choices are: • GigabitEthernet: GigabitEthernet/IEEE 802.3 interfaces. • TenGigE: TenGigabitEthernet/IEEE 802.3 interfaces.

	Command or Action	Purpose
Step 7	**neighbor** *ip-address* **pw-id** *pseudowire-id* **Example:** RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)# neighbor 2.2.2.2 pw-id 2000	Configures the pseudowire segment for the cross-connect. Optionally, you can disable the control word or set the transport-type to "Ethernet" or "vlan".
Step 8	**end** or **commit** **Example:** RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)# end or RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)# commit	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Inter-AS

The Inter-AS configuration procedure is identical to the L2VPN cross-connect configuration tasks (see "Configuring Static Point-to-Point Cross-Connects" section on page MPC-29 and "Configuring Dynamic Point-to-Point Cross-Connects" section on page MPC-31) except that the remote PE IP address used by the cross-connect configuration is now reachable through iBGP peering.

✎ **Note** You must be knowledgeable about IBGP, EBGP, and ASBR terminology and configurations to complete this configuration.

Configuring L2VPN Quality of Service

This section describes how to configure L2VPN quality of service (QoS) in port mode, VLAN mode, Frame Relay and ATM sub-interfaces.

Restrictions

The **l2transport** command cannot be used with any IP address, L3, or CDP configuration.

Configuring an L2VPN Quality of Service Policy in Port Mode

This procedure describes how to configure an L2VPN QoS policy in port mode.

✎
Note In port mode, the interface name format does not include a subinterface number; for example, GigabitEthernet0/1/0/1.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id.subinterface* **l2transport**
3. **service-policy** *[***input** | **output***] [policy-map-name]*
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`RP/0/RP0/CPU0:router# configure`
		Enters the configuration mode.

Step 2	`interface` *type interface-path-id.subinterface* `l2transport`
	
	Example:
	`RP/0/RP0/CPU0:router(config)# interface`
	`GigabitEthernet0/0/0/0.1`
		Configures an interface or connection for L2 switching and specifies the interface attachment circuit.

	Command or Action	Purpose
Step 3	**service-policy** *[***input** *|* **output***]* *[policy-map-name]* **Example:** `RP/0/RP0/CPU0:router(config-if)# service-policy input servpol1`	Attaches a QoS policy to an input or output interface to be used as the service policy for that interface.

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

Configuring an L2VPN Quality of Service Policy in VLAN Mode

This procedure describes how to configure a L2VPN QoS policy in VLAN mode.

Note In VLAN mode, the interface name must include a subinterface; for example, GigabitEthernet0/1/0/1.1; and the l2transport command must follow the interface type on the same CLI line (for example, "interface GigabitEthernet0/0/0/0.1 l2transport").

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id.subinterface* **l2transport**
3. **service-policy** [**input** | **output**] [*policy-map-name*]
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** RP/0/RP0/CPU0:router# configure	Enters the configuration mode.
Step 2	**interface** *type interface-path-id.subinterface* **l2transport** **Example:** RP/0/RP0/CPU0:router(config)# interface GigabitEthernet0/0/0/0.1 l2transport	Configures an interface or connection for L2 switching. **Note** In VLAN Mode, you must enter the **l2transport** keyword on the same line as the interface.
Step 3	**service-policy** [**input** | **output**] [*policy-map-name*] **Example:** RP/0/RP0/CPU0:router(config-if)# service-policy input servpol1	Attaches a QoS policy to an input or output interface to be used as the service policy for that interface.

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

Configuring an L2VPN Quality of Service Policy in Frame Relay Mode

This procedure describes how to configure a L2VPN QoS policy in Frame Relay mode.

SUMMARY STEPS

1. **configure**

2. **class-map match-any** [*new class map name]*

3. **match frame-relay dlci** [*dlci number*]

4. **end**
   or
   commit

DETAILED STEPS

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

Step 2	**class-map match any** *new class name*
	
	Example:
	`RP/0/0/CPU0:router(config-cmap)# class-map`
	`match-any A`
		Matches the class map type to a new class map.

	Command or Action	Purpose
Step 3	**match frame-relay dlci** *dlci number* **Example:** RP/0/0/CPU0:router(config-cmap)# match frame-relay dlci 100-200 500	Applies the quality of service on the main interface with a frame relay encapsulation type.
Step 4	**end** or **commit** **Example:** RP/0/RP0/CPU0:router(config-cmap)# end or RP/0/RP0/CPU0:router(config-cmap)# commit	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: — Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. — Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. — Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Preferred Tunnel Path

This procedure describes how to configure a preferred tunnel path.

Note The tunnel used for the preferred path configuration is an MPLS Traffic Engineering (MPLS-TE) tunnel.

SUMMARY STEPS

1. **configure**

2. **l2vpn**

3. **pw-class** {*name*}

4. **encapsulation mpls**

5. **preferred-path** {**interface**} {**tunnel-te** *value*} [**fallback disable**]

6. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** RP/0/RP0/CPU0:router# configure	Enters the configuration mode.
Step 2	**l2vpn** **Example:** RP/0/RP0/CPU0:router(config)# l2vpn	Enters L2VPN configuration mode.
Step 3	**pw-class** {*name*} **Example:** RP/0/RP0/CPU0:router(config-l2vpn)# pw-class path1	Configures the pseudowire class name.
Step 4	**encapsulation mpls** **Example:** RP/0/RP0/CPU0:router(config-l2vpn-pwc)# encapsulation mpls	Configures the pseudowire encapsulation to MPLS.

	Command or Action	Purpose
Step 5	**preferred-path** {**interface**} {**tunnel-te** *value*} [**fallback disable**] **Example:** RP/0/RP0/CPU0:router(config-l2vpn-pwc-encap-mpls)# preferred-path interface tunnel 11 fallback disable	Configures preferred path tunnel settings. If the fallback disable configuration is used and once the TE tunnel is configured as the preferred path goes down, the corresponding pseudowire can also go down.
Step 6	**end** or **commit** **Example:** RP/0/RP0/CPU0:router(config-l2vpn-pwc-encap-mpls)# end or RP/0/RP0/CPU0:router(config-l2vpn-pwc-encap-mpls-if)# commit	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring AToM IP Interworking

To configure AToM IP interworking, you need to configure attachment circuits (AC), pseudowire class, and cross connects.

- Configuring Ethernet ACs for AToM IP Interworking, page VPC-41
- Configuring Frame Relay ACs for AToM IP Interworking, page VPC-42
- Configuring ATM AAL5 ACs for AToM IP Interworking, page VPC-44

Configuring Ethernet ACs for AToM IP Interworking

Perform this task to configure an Ethernet AC for AToM IP Interworking.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **l2transport**
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`RP/0/0/CPU0:router# configure`
		Enters global configuration mode.

Step 2	`interface` *type interface-path-id* **Example:** `RP/0/0/CPU0:router(config)# interface ethernet 0/0/0/0`	Configures the Ethernet interface.

	Command or Action	Purpose
Step 3	**l2transport** **Example:** RP/0/0/CPU0:router(config-if)# l2transport	Configures the Layer 2 Transport type for the AC.

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

Configuring Frame Relay ACs for AToM IP Interworking

Perform this task to configure a Frame Relay AC for AToM IP Interworking.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **encapsulation frame-relay** *frame-relay networks*
4. **frame-relay [intf-type] dce**
5. **interface** *type interface-path-id* **l2transport**
6. **pvc** *number*
7. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** RP/0/0/CPU0:router# configure	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** RP/0/0/CPU0:router(config)# interface POS 0/2/0/1	Configures the Layer 2 transport sub-interface.
Step 3	**encapsulation frame-relay** *frame-relay networks* **Example:** RP/0/0/CPU0:router(config-if)# encapsulation frame-relay	Encapsulates the Frame Relay network using RFC1490 or RFC2427 encapsulation.
Step 4	**frame-relay [intf-type] dce** **Example:** RP/0/RP0/CPU0:router(config-if)# frame	Configures Frame Relay interface type based on the DCE mode.
Step 5	**interface** *type interface-path-id* **l2transport** **Example:** RP/0/0/CPU0:router(config)# interface POS 0/2/0/1.200 l2transport	Configures the Layer 2 transport sub-interface.

	Command or Action	Purpose
Step 6	**pvc** *number* **Example:** RP/0/0/CPU0:router(config-subif)# pvc 20	Configures a virtual circuit.
Step 7	**end** or **commit** **Example:** RP/0/0/CPU0:router(config-if)# end or RP/0/0/CPU0:router(config-if)# commit	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring ATM AAL5 ACs for AToM IP Interworking

Perform this task to configure an ATM AAL5 AC for AToM IP Interworking.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id* **l2transport**
3. **pvc** *number*
4. **encapsulation {aal5mux} {ipv4}**
5. **Repeat steps 1 through 3**
6. **encapsulation {aal5snap}**
7. **end**
   or
   commit

DETAILED STEPS

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

Step 2	**interface** *type interface-path-id* **l2transport**
	
	Example:
	`RP/0/0/CPU0:router(config)# interface ATM`
	`0/2/0/1.200 l2transport`
		Configures the Layer 2 transport sub-interface.

Step 3	**pvc** *number* **Example:** `RP/0/0/CPU0:router(config-subif)# pvc 2/200`	Configures a virtual circuit.

Step 4	**encapsulation {aal5mux} {ipv4}**
	
	Example:
	`RP/0/0/CPU0:router(config-atm-l2transport-pvc)#`
	`encapsulation aal5mux ipv4`
		Configures the AAL5 MUX ATM encapsulation over an IPv4 network.

Step 5	**encapsulation {aal5snap}**
	
	Example:
	`RP/0/0/CPU0:router(config-atm-l2transport-pvc)#`
	`encapsulation aal5snap`
		Configures the AAL5 SNAP ATM encapsulation.

Step 6	**end**
	or
	commit
	
	Example:
	`RP/0/0/CPU0:router(config-atm-l2transport-pvc)#`
	` end`
	or
	`RP/0/0/CPU0:router(config-atm-l2transport-pvc)#`
	`commit`
		Saves configuration changes.
		
		• When you issue the **end** command, the system prompts you to commit changes:
		
		`Uncommitted changes found, commit them before`
		`exiting(yes/no/cancel)?`
		`[cancel]:`
		
		  – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		
		  – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		
		  – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuration Examples for L2VPN

In the following example, two traffic classes are created and their match criteria are defined. For the first traffic class called class1, ACL 101 is used as the match criterion. For the second traffic class called class2, ACL 102 is used as the match criterion. Packets are checked against the contents of these ACLs to determine if they belong to the class.

This section includes the following configuration examples:

L2VPN Interface Configuration: Example

The following example shows how to configure an L2VPN interface:

```
configure
 interface GigabitEthernet0/0/0/0.1 l2transport
 dot1q vlan 1
 end
```

Point-to-Point Cross-connect Configuration: Examples

This section includes configuration examples for both static and dynamic point-to-point cross-connects.

Static Configuration

The following example shows how to configure a static point-to-point cross-connect:

```
configure
 l2vpn
  xconnect group vlan_grp_1
   p2p vlan1
   interworking ipv4
   interface GigabitEthernet0/0/0/0.1
   neighbor 2.2.2.2 pw-id 2000
    mpls static label local 699 remote 890
    commit
```

Dynamic Configuration

The following example shows how to configure a dynamic point-to-point cross-connect:

```
configure
 l2vpn
  xconnect group vlan_grp_1
   p2p vlan1
   interworking ipv4
   interface GigabitEthernet0/0/0/0.1
   neighbor 2.2.1.1 pw-id 1commit
```

Inter-AS: Example

The following example shows how to set up an AC to AC cross-connect from AC1 to AC2:

```
router-id Loopback0

interface Loopback0
 ipv4 address 127.0.0.1 255.255.255.0
!
interface GigabitEthernet0/1/0/0.1 l2transport dot1q vlan 1!
!
interface GigabitEthernet0/0/0/3
 ipv4 address 127.0.0.1 255.255.255.0
 keepalive disable
!
interface GigabitEthernet0/0/0/4
 ipv4 address 127.0.0.1 255.255.255.0
 keepalive disable
!
```

```
router ospf 100
 log adjacency changes detail
 area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/3
  !
  interface GigabitEthernet0/0/0/4
  !
 !
!
router bgp 100
 address-family ipv4 unicast
  allocate-label all
 !
 neighbor 40.0.0.5
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family ipv4 labeled-unicast
  !
 !
!
l2vpn
 xconnect group xc1
  p2p ac2ac1
   interface GigabitEthernet0/1/0/0.1
   neighbor 20.0.0.5 pw-id 101
  !
  p2p ac2ac2
   interface GigabitEthernet0/1/0/0.2
   neighbor 20.0.0.5 pw-id 102
  !
  p2p ac2ac3
   interface GigabitEthernet0/1/0/0.3
   neighbor 20.0.0.5 pw-id 103
  !
  p2p ac2ac4
   interface GigabitEthernet0/1/0/0.4
   neighbor 20.0.0.5 pw-id 104
  !
  p2p ac2ac5
   interface GigabitEthernet0/1/0/0.5
   neighbor 20.0.0.5 pw-id 105
  !
  p2p ac2ac6
   interface GigabitEthernet0/1/0/0.6
   neighbor 20.0.0.5 pw-id 106
  !
  p2p ac2ac7
   interface GigabitEthernet0/1/0/0.7
   neighbor 20.0.0.5 pw-id 107
  !
  p2p ac2ac8
   interface GigabitEthernet0/1/0/0.8
   neighbor 20.0.0.5 pw-id 108
  !
  p2p ac2ac9
   interface GigabitEthernet0/1/0/0.9
   neighbor 20.0.0.5 pw-id 109
  !
  p2p ac2ac10
   interface GigabitEthernet0/1/0/0.10
```

```
    neighbor 20.0.0.5 pw-id 110
   !
  !
!
mpls ldp
 router-id Loopback0
 log
  neighbor
 !
 interface GigabitEthernet0/0/0/3
 !
 interface GigabitEthernet0/0/0/4
 !
!
end
```

L2VPN Quality of Service: Example

The following example shows how to attach a service-policy to an L2 interface in port mode:

```
configure
  interface GigabitEthernet 0/0/0/0
  l2transport
  service-policy [input | output] [policy-map-name]
commit
```

Preferred Path: Example

The following example shows how to configure preferred tunnel path:

```
configure
 l2vpn
 pw-class path1
  encapsulation mpls
   preferred-path interface tunnel value fallback disable
```

AToM IP Interworking: Examples

This section includes configuration examples for all supported AC modes in AToM IP Interworking.

Ethernet

```
interface GigabitEthernet0/0/0/2
  l2transport
!
interface GigabitEthernet0/0/0/3.1 l2transport
  dot1q vlan 1
!
interface GigabitEthernet0/0/0/3.2 l2transport
dot1q vlan 2 2
```

Frame Relay

```
interface POS0/2/0/1
mtu 1500
encapsulation frame-relay
frame-relay intf-type dce
!
interface POS0/2/0/1.20 l2transport
pvc 20
!
```

ATM AAL5

```
interface ATM0/3/0/1.200 l2transport
pvc 20/200
encapsulation aal5mux ipv4
!
interface ATM0/3/0/1.300 l2transport
pvc 30/300
encapsulation aal5snap
!
interface ATM0/3/0/1.300 l2transport
pvc 30/400
encapsulation aal5nlpid
```

AToM Cross Connect Configuration: Example

This section includes configuration examples for all supported AToM Cross Connects.

```
l2vpn
pseudowire-class ipiw
  encapsulation mpls
!
xconnect group port
  p2p port1
    interface GigabitEthernet0/0/0/2
    neighbor 11.11.11.11 pw-id 300 pw-class ipiw
  !
!
xconnect group vlan
  p2p vlan1
    interface GigabitEthernet0/0/0/3.1
    neighbor 11.11.11.11 pw-id 400 pw-class ipiw
  !
!
xconnect group frame-relay
  p2p frame1
    interface POS0/2/0/1.20
    neighbor 11.11.11.11 pw-id 600 pw-class ipiw
  !
!
xconnect group atm
  p2p atm1
    interface ATM0/3/0/1.200
    neighbor 11.11.11.11 pw-id 700 pw-class ipiw
  !
  p2p atm2
    interface ATM0/3/0/1.300
    neighbor 11.11.11.11 pw-id 800 pw-class ipiw
```

Configuring L2VPN over GRE Tunnels: Example

The following example shows how to configure L2VPN over GRE tunnels:

```
interface tunnel-ip101
 ipv4 address 150.10.1.204 255.255.255.0
 ipv6 address 150:10:1::204/64
 tunnel mode gre ipv4
 tunnel source Loopback1
 tunnel destination 100.1.1.202

router ospf 1
 router-id 100.0.1.204
 cost 1
 router-id Loopback0
 area 1
  interface Loopback0
  !
  interface tunnel-ip101

mpls ldp
 router-id 100.0.1.204
 interface tunnel-ip101

l2vpn
 xconnect group pe2
  p2p 2001
    interface GigabitEthernet0/2/0/0.2001
    neighbor 100.0.1.202 pw-id 2001
```

Additional References

For additional information related to implementing MPLS Layer 2 VPN, refer to the following references:

Related Documents

Related Topic	Document Title
Cisco IOS XR L2VPN command reference document	*MPLS Virtual Private Network Commands on Cisco IOS XR Software* module in *Cisco IOS XR MPLS Command Reference*
MPLS VPN-related commands	*MPLS Virtual Private Network Commands on Cisco IOS XR Software* module in *Cisco IOS XR MPLS Command Reference*
MPLS Layer 2 VPNs	*Implementing MPLS Layer 2 VPNs on Cisco IOS XR Software* module in *Cisco IOS XR MPLS Configuration Guide*
MPLS Layer 3 VPNs	*Implementing MPLS Layer 3 VPNs on Cisco IOS XR Software* module in *Cisco IOS XR MPLS Configuration Guide*
MPLS VPNs over IP Tunnels	*MPLS VPNs over IP Tunnels on Cisco IOS XR Software* module in *Cisco IOS XR MPLS Configuration Guide*

Related Topic	Document Title
Cisco CRS router getting started material	*Cisco IOS XR Getting Started Guide*
Information about user groups and task IDs	*Configuring AAA Services on Cisco IOS XR Software* module of *Cisco IOS XR System Security Configuration Guide*

Standards

Standards[1]	Title
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	—

1. Not all supported standards are listed.

MIBs

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

RFCs

RFCs	Title
RFC 3931	*Layer Two Tunneling Protocol - Version 3 (L2TPv3)*
RFC 4447	*Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*, April 2006
RFC 4448	*Encapsulation Methods for Transport of Ethernet over MPLS Networks*, April 2006

Technical Assistance

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