Configuring Any Transport over MPLS

This chapter describes how to configure the Any Transport over MPLS (AToM) feature.

This chapter includes the following sections:

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your 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 "New and Changed Information" chapter or the Feature History table below.

Information About Any Transport over MPLS

This section includes the following topics:

- Equal Cost Multiple Paths on PWE Label, page 27-486

Any Transport over MPLS

Any Transport over MPLS (AToM) accommodates different types of Layer 2 packets, including Ethernet and VLAN, to enable the service provider to transport different 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. 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 successful transmission of the Layer 2 frames between PE devices is due to the configuration of the PE devices. You can set up the connection, called a pseudowire, between the routers and specify the following information on each PE device:

- The type of Layer 2 data to be transported across the pseudowire, such as Ethernet or VLAN.
- The IP address of the loopback interface of the peer PE device, which enables PE devices to communicate.
- A unique combination of peer PE IP address and virtual circuit (VC) ID that identifies the pseudowire.

AToM encapsulates Layer 2 frames at the ingress provider edge (PE) and sends them to a corresponding PE at the other end of a pseudowire. The egress PE removes the encapsulation and sends out the Layer 2 frame.

Ethernet over MPLS

Any Transport over MPLS (AToM) supports Ethernet over MPLS (EoMPLS) in two modes: VLAN and port mode.

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

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 frame check sequence (FCS) is transported as a single packet. Each interface is associated with one unique pseudowire VC label.

Ethernet Remote Port Shutdown

Ethernet remote port shutdown allows a service provider edge (PE) device on the local end of an Ethernet over MPLS (EoMPLS) pseudowire to detect a remote link failure and cause the shutdown of the Ethernet port on the local customer edge (CE) device. Because the Ethernet port on the local CE device is shut down, the device does not lose data by continuously sending traffic to the failed remote link. This process is beneficial if the link is configured as a static IP route.

Estimating Packet Sizes

The following calculation helps you to determine the size of the packets that travel through the core network. You must set the maximum transmission unit (MTU) on the core-facing interfaces of the provider (P) and provider edge (PE) devices to accommodate packets of the calculated 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)), where the following definitions apply:

- The edge MTU is the MTU for customer-facing devices.
- The Transport header depends on the transport type. The table below lists the specific sizes of the headers.

Transport Type	Packet Size
Ethernet VLAN	18 bytes
Ethernet port	14 bytes

- The AToM header is 4 bytes (control word).
- 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 devices that do not have a P router between them.
  - If the Label Distribution Protocol (LDP) is used in the MPLS network, the label stack size is two (the LDP label and the VC label).
  - If a traffic engineering (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 Fast Reroute (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, 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.

Note For more information about establishing nondirectly connected MPLS LDP sessions, see the "Configuring MPLS Label Distribution Protocol" chapter.

Applying the following assumptions and using the formula: Edge MTU + Transport header + AToM header + (MPLS label stack * MPLS label) = Core MTU, or 1500 + 18 + 0 + (2 * 4) = 1526, you must configure the P and PE devices in the core to accept packets of 1526 bytes.

- The edge MTU is 1500 bytes.
- The transport type is Ethernet VLAN which is 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.

Layer 2 VPN Internetworking

Layer 2 transport over Multiprotocol Label Switching (MPLS) already exists for like-to-like attachment circuits, such as Ethernet-to-Ethernet. Layer 2 Virtual Private Network (L2VPN) internetworking builds on this functionality by allowing disparate attachment circuits to be connected. The internetworking function facilitates the translation between the different Layer 2 encapsulations.

The EoMPLS L2VPN Internetworking feature supports Ethernet and VLAN attachment circuits over MPLS. The features and restrictions for like-to-like functionality also apply to L2VPN internetworking.

Quality of Service Features Supported in AToM

The table below lists the Quality of Service (QoS) features supported in AToM.

Table 27-1 QoS Features Supported in AToM

QoS Feature	EoMPLS
Service policy	Can be applied to Ethernet Virtual Circuits (EVCs) and switchport interfaces
Classification	Supports the commands for matching the following: • Class of service (CoS) on interfaces and subinterfaces • MPLS experimental topmost on interfaces and subinterfaces • QoS groups on interfaces (output policy)

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

Equal Cost Multiple Paths on PWE Label

Equal Cost Multiple Paths (ECMPs) are available between the ingress and egress devices. However, a pseudowire is transported over a single network path to retain the characteristics of the emulated service over the pseudowire.

In the network core, load balancing is performed by checking the first nibble in the frame, after the MPLS label stack. If the destination MAC address (DMAC) starts with 4 or 6, it selects a different link in the core. To avoid a different link and preserve order of frames, a control word is added to the frame transmitted over the pseudowire emulation (PWE) label.

Licensing Requirements for Any Transport over MPLS

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	Layer 2 MVPN requires an MPLS license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Guidelines and Limitations for Any Transport over MPLS

Any Transport over MPLS (AToM) has the following configuration guidelines and limitations:

- 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) has the following guidelines and limitations:

- EoMPLS 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 customer edge (CE) devices.
- 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.
- A switch can act as the terminating provider edge (T-PE) router and peer with the subscriber provider edge (S-PE) router. But a switch cannot act as an S-PE router.
- Although you can set the MPLS maximum transmission unit (MTU) to a value less than the interface MTU, you must set the MPLS MTU to a value greater 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, you can set the maximum MPLS MTU as high as the interface MTU. For example, if the interface MTU is set to 1600 bytes, you can set the MPLS MTU to a maximum of 1600 bytes. If you set the MPLS MTU to a value 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.

Per-interface Ethernet over MPLS (EoMPLS) has the following guidelines and restrictions:

- The Virtual Private LAN Service (VPLS) feature does not support MTU values in pseudowire interface configuration mode.
- The device uses an MTU validation process for remote virtual circuits (VCs) established through LDP, which compares the MTU value configured in pseudowire interface configuration mode to the MTU value of the remote customer interface. If an MTU value has not been configured in pseudowire interface configuration mode, the validation process compares the MTU value of the local customer interface to the MTU value of the remote, either explicitly configured or inherited from the underlying interface or subinterface.
- When you configure the MTU value in pseudowire interface configuration mode, the specified MTU value is not enforced by the dataplane. The dataplane enforces the MTU values of the interface (port mode) or subinterface (VLAN mode).
- Ensure that the interface MTU is larger than the MTU value configured in pseudowire interface configuration mode. If the MTU value of the customer-facing subinterface is larger than the MTU value of the core-facing interface, traffic might not be able to travel across the pseudowire.

Configuring Any Transport over MPLS

This section includes the following topics:

Configuring a Pseudowire

BEFORE YOU BEGIN

Ensure that you configured the EFP (service instance) for EoMPLS. For information, see the "Configuring Ethernet over MPLS" chapter.

SUMMARY STEPS

1. **configure terminal**
2. **port-profile type pseudowire** *profile-name*
3. **encapsulation mpls**
4. [**no**] **interface pseudowire** *pw-id*
5. (Optional) **control-word**
6. **inherit port-profile** *profile-name*
7. **neighbor** *peer-ip-address vc-id*
8. (Optional) **copy running-config start-up config**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**port-profile type pseudowire** *profile-name*
	
	Example:
	`switch(config)# port-profile type pseudowire AToM`
	`switch(config-if-prof)#`
		Enters interface port-profile configuration mode and configures a pseudowire port profile.

Step 3	**encapsulation mpls** **Example:** `switch(config-if-prof)# encapsulation mpls`	Specifies MPLS encapsulation for this profile.

Step 4	[**no**] **interface pseudowire** *pw-id*
	
	Example:
	`switch(config-prof)# interface pseudowire 12`
	`switch(config-if-pseudowire)#`
		Enters interface pseudowire configuration mode and configures a static pseudowire logical interface.
		• The *pw-id* argument is a unique per-interface identifier for this pseudowire. The range is from 1 to 200000. The range for a static pseudowire is from 1 to 8192.
		Note You can use the **no** form of this command to delete the pseudowire interface and the associated configuration.

Step 5	**control-word** **Example:** `switch(config-if-pseudowire)# control-word`	(Optional) Enables the control word for this interface. • If you do not enable a control word, autosense is the default mode for the control word.

	Command	Purpose
Step 6	**inherit port-profile** *profile-name* **Example:** `switch(config-if-pseudowire)# inherit port-profile AToM`	Applies a port profile to this interface.
Step 7	**neighbor** *peer-ip-address vc-id* **Example:** `switch(config-if-pseudowire)# neighbor 10.2.2.1 1`	Configures a emulated virtual circuit for this interface. • The combination of the *peer-ip-address* and *vc-id* arguments must be unique on a device. • The peer IP address is the address of the provider edge (PE) peer. • The *vc-id* argument is an identifier for the virtual circuit between devices. The valid range is from 1 to 4294967295.
Step 8	**copy running-config startup-config** **Example:** `switch(config-xconnect)# copy running-config startup-config`	(Optional) Saves this configuration change.

Configuring Ethernet Remote Port Shutdown (optional)

The Remote Ethernet Port Shutdown feature is enabled by default when an image with the feature supported is loaded on the device.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **l2vpn xconnect context** *context-name*
3. [**no**] **remote failure notification**
4. (Optional) **copy running-config start-up config**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **l2vpn xconnect context** *context-name*
	
	Example:
	`switch(config)# l2vpn context cxt1`
	`switch(config-xconnect)#`
		Enters Xconnect configuration mode and establishes a Layer 2 VPN (L2VPN) context for identifying the two members in a VPWS, multisegment pseudowire, or local connect service.
		• The *context-name* argument is a unique per-interface identifier for this context. The maximum range is 100 alphanumeric, case-sensitive characters.
		Note You can use the **no** form of this command to delete the context and the associated configuration.

Step 3	[**no**] **remote failure notification**
	
	Example:
	`switch(config-xconnect)#`
	`remote failure notification`
		Enables AToM MPLS remote link failure notification and shutdown.
		Note You can use the **no** form of this command to disable this feature.

Step 4	**copy running-config startup-config** **Example:** `switch(config-xconnect)# copy running-config startup-config`	(Optional) Saves this configuration change.

Configuring Ethernet over MPLS in VLAN Mode

You can configure EoMPLS (VLAN mode) on the subinterfaces.

BEFORE YOU BEGIN

- Ensure that you configured the EFP (service instance) for EoMPLS. For information, see the "Configuring Ethernet over MPLS" chapter.
- Before configuring Ethernet over MPLS (EoMPLS) in VLAN mode, you must configure EoMPLS on the subinterfaces.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/subslot/port*[**.***subinterface*]
3. **encapsulation dot1q** *vlan-id*
4. [**no**] **l2vpn context** *context-name*

5. (Optional) **internetworking** {**ethernet** | **vlan**}
6. [**no**] **member** *interface-type slot/port* [**service-instance** *service-instance-id*] [**group** *group-name*] [**priority** *number*]
7. [**no**] **member pseudowire** *pw-id* [**group** *name*] [**priority** *number*]
8. (Optional) **copy running-config start-up config**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot/subslot/port*[**.***subinterface*]
	
	Example:
	`switch(config)# interface ethernet 4/0/0.1`
	`switch(config-if)#`
		Enters interface configuration mode.
		• Ensure the subinterface on the adjoining CE router is on the same VLAN as this PE router.

Step 3	**encapsulation dot1q** *vlan-id* **Example:** switch(config-if)# encapsulation dot1q 100	Configures the matching criteria for mapping dot1q frames on an ingress interface to this interface. • The valid range for the *vlan-id* argument is from 2 to 967. • The subinterfaces between the CE and PE routers that are running EoMPLS must be in the same subnet. All other subinterfaces and backbone devices do not ned to be in the same subnet.

Step 4	[**no**] **l2vpn xconnect context** *context-name*
	
	Example:
	`switch(config-if)# l2vpn context cxt1`
	`switch(config-xconnect)#`
		Enters XConnect configuration mode and establishes a Layer 2 VPN (L2VPN) context for identifying the two members in a VPWS, multisegment pseudowire, or local connect service.
		• The *context-name* argument is a unique per-interface identifier for this context. The maximum range is 100 alphanumeric, case-sensitive characters.
		Note You can use the **no** form of this command to delete the context and the associated configuration.

	Command	Purpose
Step 5	**internetworking** {**ethernet** | **vlan**} **Example:** `switch(config-xconnect)# internetworking ethernet`	(Optional) Specifies the type of pseudowire and the type of traffic that can flow across it. • This command is required only if you are configuring a connection between two disparate attachment circuits. • The internetworking type on a provider edge (PE) device must match the internetworking type on its peer PE device. • 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. • The **vlan** keyword allows the VLAN ID to be included as part of the Ethernet frame.
Step 6	[**no**] **member** *interface-type slot*/*port* [**service-instance** *service-instance-id*] [**group** *group-name*] [**priority** *number*] **Example:** `switch(config-xconnect)# member ethernet 0/0/0.1 service-instance 300`	Adds an active Ethernet AC, with or without an Ethernet Flow Point (EFP), to the context. • The *service-instance-id* argument is a unique per-interface identifier for the EFP. The valid range is from 1 to 4000. The range might be restricted due to resource constraints. • (Optional) The **group** *group-name* keyword and argument combination specifies to which of the redundant groups the member belongs. This configuration is required if the member is backed up by one or more other group members in order to identify to which redundant group each member belongs. • (Optional) The **priority** *number* keyword and argument combination specifies the priority of the backup pseudowire in instances where multiple backup pseudowires exist. The range is from 1 to 10, with 1 being the highest priority. The default is 0 and is higher than 1. • You can use the **no** form of this command to delete the specified member configuration.

	Command	Purpose
Step 7	[**no**] **member pseudowire** pw-id [**group** group-*name*] [**priority** *number*] **Example:** `switch(config-xconnect)# member pseudowire 12 group core-side priority 1`	Adds an active pseudowire to the context. • The *pw-id* argument is a unique per-interface identifier for this pseudowire. The range is from 1 to 200000. The range for a static pseudowire is from 1 to 8192. • (Optional) The **group** *group-name* keyword and argument combination specifies to which of the redundant groups the member belongs. This configuration is required if the member is backed up by one or more other group members in order to identify to which redundant group each member belongs. • (Optional) The **priority** *number* keyword and argument combination specifies the priority of the backup pseudowire in instances where multiple backup pseudowires exist. The range is from 1 to 10, with 1 being the highest priority. The default is 0 and is higher than 1. • You can use the **no** form of this command to delete the specified member configuration.
Step 8	**copy running-config startup-config** **Example:** `switch(config-xconnect)# copy running-config startup-config`	(Optional) Saves this configuration change.

Configuring Ethernet over MPLS in Port Mode

Perform this task to configure EoMPLS (port mode) on the subinterfaces.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/subslot/port*[**.***subinterface*]
3. **l2vpn xconnect context** *context-name*
4. [**no**] **member** *interface-type slot/port* [**service-instance** *service-instance-id*] [**group** *group-name*] [**priority** *number*]
5. [**no**] **member pseudowire** *pw-id* [**group** *name*] [**priority** *number*]
6. (Optional) **copy running-config start-up config**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot/subslot/port*[**.***subinterface*]
	
	Example:
	`switch(config)# interface ethernet 4/0/0`
	`switch(config-if)#`
		Enters interface configuration mode.
		• Ensure the subinterface on the adjoining CE router is on the same VLAN as this PE router.

Step 3	[**no**] **l2vpn xconnect context** *context-name*
	
	Example:
	`switch(config-if)# l2vpn context cxt1`
	`switch(config-xconnect)#`
		Enters XConnect configuration mode and establishes a Layer 2 VPN (L2VPN) context for identifying the two members in a VPWS, multisegment pseudowire, or local connect service.
		• The *context-name* argument is a unique per-interface identifier for this context. The maximum range is 100 alphanumeric, case-sensitive characters.
		Note You can use the **no** form of this command to delete the context and the associated configuration.

Step 4	[**no**] **member** *interface-type slot/port* [**service-instance** *service-instance-id*] [**group** *group-name*] [**priority** *number*] **Example:** `switch(config-xconnect)# member ethernet 0/0`	Adds an active Ethernet AC, with or without an Ethernet Flow Point (EFP), to the context. • The *service-instance-id* argument is a unique per-interface identifier for the EFP. The valid range is from 1 to 4000. The range might be restricted due to resource constraints. • (Optional) The **group** keyword specifies which of redundant groups the member belongs. This must be configured if the member is backed up by one or more other group members in order to identify to which redundant group each member belongs. • (Optional) The **priority** *number* keyword and argument combination specifies the priority of the backup pseudowire in instances where multiple backup pseudowires exist. The range is from 1 to 10, with 1 being the highest priority. The default is 0 and is higher than 1. • You can use the **no** form of this command to delete the specified member configuration.

	Command	Purpose
Step 5	[**no**] **member pseudowire** pw-id [**group** *name*] [**priority** *number*] **Example:** `switch(config-xconnect)# member pseudowire 12`	Adds an active pseudowire to the context. • The *pw-id* argument is a unique per-interface identifier for this pseudowire. The range is from 1 to 200000. The range for a static pseudowire is from 1 to 8192. • (Optional) The **group** keyword specifies which of redundant groups the member belongs. This must be configured if the member is backed up by one or more other group members in order to identify to which redundant group each member belongs. • (Optional) The **priority** *number* keyword and argument combination specifies the priority of the backup pseudowire in instances where multiple backup pseudowires exist. The range is from 1 to 10, with 1 being the highest priority. The default is 0 and is higher than 1. • You can use the **no** form of this command to delete the specified member configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config-xconnect)# copy running-config startup-config`	(Optional) Saves this configuration change.

Configuring Per-Subinterface MTU for Ethernet over MPLS

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **mtu** *mtu-value*
4. **encapsulation dot1q** *vlan-id*
5. [**no**] **l2vpn context** *context-name* **encapsulation mpls**
6. **mtu** *mtu-value*
7. (Optional) **copy running-config start-up config**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot/port*
	
	Example:
	`switch(config)# interface ethernet 3/1`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**mtu** *mtu-value* **Example:** `switch(config-if)# mtu 2000`	Configures the maximum transmission unit (MTU) size, in bytes, for this interface. • The valid range for the *mtu-value* argument is 576 to 9216. The default is 1500.
Step 4	**encapsulation dot1q** *vlan-id* **Example:** switch(config-if)# encapsulation dot1q 100	Configures the matching criteria for mapping dot1q frames on an ingress interface to this EFP. • The valid range for the *vlan-id* argument is from 2 to 967. • The subinterfaces between the CE and PE routers that are running EoMPLS must be in the same subnet. All other subinterfaces and backbone devices do not need to be in the same subnet.

Step 5	[**no**] **l2vpn context** *context-name* **encapsulation mpls**
	
	Example:
	`switch(config-if)# l2vpn context cxt1 encapsulation mpls`
	`switch(config-xconnect)#`
		Enters Xconnect configuration mode and establishes a Layer 2 VPN (L2VPN) context for identifying the two members in a VPWS, multisegment pseudowire, or local connect service.
		• The *context-name* argument is a unique per-interface identifier for this context. The maximum range is 100 alphanumeric, case-sensitive characters.
		• The **encapsulation** and **mpls** keywords specify MPLS encapsulation for this context.
		Note You can use the **no** form of this command to delete the context and the associated configuration.

Step 6	**mtu** *mtu-value* **Example:** `switch(config-xconnect)# mtu 1400`	Configures the maximum transmission unit (MTU) size, in bytes, for this context. • The valid range for the *mtu-value* argument is 576 to 9216. The default is 1500.
Step 7	**copy running-config startup-config** **Example:** `switch(config-xconnect)# copy running-config startup-config`	(Optional) Saves this configuration change.

Verifying Any Transport over MPLS

To verify configuration information, perform one of the following tasks:

Command	Purpose
show l2vpn atom vc detail	Displays detailed information about Any Transport over MPLS (AToM) virtual circuits (VCs) and static pseudowires that have been enabled to route Layer 2 packets on a device.
show l2vpn mpls transport binding	Displays the MTU values assigned to the local and remote interfaces.

Configuration Examples for Any Transport over MPLS

This section includes the following topics:

Example: Remote Ethernet Port Shutdown

The following example shows how to enable a remote Ethernet port shutdown:

```
interface pseudowire 100
    encapsulation mpls
    neighbor 10.1.1.1 1
!
l2vpn xconnect context con1
    remote link failure notification
```

The following example shows how to disable a remote Ethernet port shutdown:

```
interface GigabitEthernet1/0/0
    interface pseudowire 100
    encapsulation mpls
    neighbor 10.1.1.1 1
!
l2vpn xconnect context con1
    no remote link failure notification
```

Example: Configuring per-Subinterface MTU for Ethernet over MPLS

This example shows a configuration that enables matching MTU values between VC endpoints. PE1 is configured in the 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.

CE1 Configuration

```
interface gigabitethernet 0/0
  mtu 1500
  no ip address
```

```
!
interface gigabitethernet 0/0.1
  encapsulation dot1Q 100
  ip address 10.181.182.1 255.255.255.0
```

PE1 Configuration

```
interface gigabitethernet 0/0
  mtu 2000
  no ip address
!
interface gigabitethernet 0/0.1
  encapsulation dot1Q 100
!
interface pseudowire 100
  neighbor 10.1.1.152 100
  encapsulation mpls
  mtu 2000
!
l2vpn xconnect context ctx1
  member gigabitethernet0/0.1
  member pseudowire 100
!
interface gigabitethernet 0/0.2
  encapsulation dot1Q 200
  ip address 10.151.100.1 255.255.255.0
  mpls ip
```

PE2 Configuration

```
interface gigabitethernet 1/0
  mtu 2000
  no ip address
!
interface gigabitethernet 1/0.2
  encapsulation dot1Q 200
  ip address 10.100.152.2 255.255.255.0
  mpls ip
!
interface fastethernet 0/0
  no ip address
!
interface fastethernet 0/0.1
  description default MTU of 1500 for FastEthernet
  encapsulation dot1Q 100
  xconnect 10.1.1.151 100 encapsulation mpls
```

CE2 Configuration

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

Example: Configuring MTU for Interworking

The following example shows an L2VPN interworking example. The PE1 device has a serial interface configured with an MTU value of 1492 bytes. The PE2 router is configured with a matching MTU of 1492 bytes, which allows the two devices to form an interworking VC. If the PE2 device was not explicitly configured with a matching MTU value, the interface would be set to 1500 bytes by default and the VC would not come up.

PE1 Configuration

```
interface Loopback0
  ip address 10.1.1.151 255.255.255.255
!
interface pseudowire100
  neighbor 10.1.1.152 100
  encapsulation mpls
  mtu 2000
  l2vpn xconnect context ctx1
  member gigabitethernet0/0
  member pseudowire 100
!
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
```

Additional References for Any Transport over MPLS

For additional information about provisioning static pseudowires for Any Transport over MPLS (AToM), see the following section:

- Related Documents, page 27-500

Related Documents

Related Topic	Document Title
Interface commands	*Cisco Nexus 7000 Series NX-OS Interfaces Command Reference*
VLAN commands	Cisco Nexus 7000 Series NX-OS Layer 2 Switching Command Reference
Ethernet over MPLS	"Configuring Ethernet over MPLS" chapter
Non directly connected MPLS LDP sessions	"Configuring the MPLS Label Distribution Protocol" chapter

Feature Information for Any Transport over MPLS

Table 27-2 lists the release history for this feature.

Table 27-2 Feature Information for Any Transport over MPLS

Feature Name	Releases	Feature Information
Any Transport over MPLS	6.2(2)	The Any Transport over MPLS (AToM) feature provides the following capabilities: • Transports data link layer (Layer2) packets over a Multiprotocol Label Switching (MPLS) backbone. • Enables 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. • Provides a common framework to encapsulate and transport Ethernet traffic over an MPLS network core.