Configuring Ethernet-over-MPLS and Pseudowire Redundancy

- Configuring Ethernet-over-MPLS, on page 1
- Configuring Pseudowire Redundancy, on page 16
- Feature History for Ethernet-over-MPLS and Pseudowire Redundancy, on page 31

Configuring Ethernet-over-MPLS

This section provides information about how to configure Ethernet over Multiprotocol Label Switching (EoMPLS).

Prerequisites for Ethernet-over-MPLS

Before you configure EoMPLS, ensure that the network is configured as follows:

- Configure IP routing in the core so that the provider edge (PE) devices can reach each other through IP.
- Configure MPLS in the core so that a label switched path (LSP) exists between the PE devices.
- Configure the **no switchport**, **no keepalive**, and **no ip address** commands before configuring Xconnect on the attachment circuit.
- For load-balancing, configuring the **port-channel load-balance** command is mandatory.
- Subinterfaces must be supported to enable EoMPLS VLAN mode.
- The **mpls ldp graceful-restart** command must be configured to enable the device to protect LDP bindings and MPLS forwarding state during a disruption in service. We recommend you to configure this command (even if you do not want to preserve the forwarding state) to avoid device failure during SSO in a high availability setup with scale configurations.

Restrictions for Ethernet-over-MPLS

The following sections list the restrictions for EoMPLS port mode and EoMPLS VLAN mode.

Restrictions for Ethernet-over-MPLS Port Mode

- Ethernet Flow Point is not supported.
- Quality of Service (QoS): Customer differentiated services code point (DSCP) re-marking is not supported with virtual private wire service (VPWS) and EoMPLS.
- Virtual Circuit Connectivity Verification (VCCV) ping with explicit null is not supported.
- Layer 2 Protocol Tunneling CLI is not supported.
- Flow-Aware Transport (FAT) Pseudowire Redundancy is supported only in Protocol-CLI mode. Supported load-balancing parameters are Source IP, Source MAC address, Destination IP, and Destination MAC address.
- MPLS QoS is supported only in pipe and uniform mode. Default mode is pipe mode.
- Both legacy Xconnect and Protocol-CLI (interface pseudowire configuration) modes are supported.
- Xconnect mode cannot be configured on SVI.
- Xconnect and MACSec cannot be configured on the same interface.
- MACSec should be configured on CE devices and Xconnect should be configured on PE devices.
- A MACSec session should be available between CE devices.
- By default, EoMPLS PW tunnels all the protocols such as Cisco Discovery Protocol and Spanning Tree Protocol (STP). EoMPLS PW cannot perform selective protocol tunneling as part of L2 Protocol Tunneling CLI.

Restrictions for EoMPLS VLAN Mode

- Virtual circuit will not work if the same interworking type is not configured on PE devices.
- Untagged traffic is not supported as incoming traffic.
- Xconnect mode cannot be enabled on Layer 2 subinterfaces because multiplexer user-network interface (MUX UNI) is not supported.
- Xconnect mode cannot be configured on subinterfaces if it is enabled on the main interface for port-to-port transport.
- FAT can be configured on Protocol CLI mode only.
- In VLAN mode EoMPLS, only those packets encrypted with the dot1q in clear by the CE device will be processed by the PE device.
- QoS: Customer DSCP Remarking is not supported with VPWS and EoMPLS.
- MPLS QoS is supported in pipe and uniform mode. Default mode is pipe mode.
- In VLAN mode EoMPLS, Cisco Discovery Protocol packets from the CE will be processed by the PE, but will not be carried over the EoMPLS virtual circuit, whereas in port mode, Cisco Discovery Protocol packets from the CE will be carried over the virtual circuit.
- Only Ethernet and VLAN interworking types are supported.
- L2 Protocol Tunneling is not supported.

Information About Ethernet-over-MPLS

EoMPLS is one of the Any Transport over MPLS (AToM) transport types. EoMPLS works by encapsulating Ethernet protocol data units (PDUs) in MPLS packets and forwarding them across the MPLS network. Each PDU is transported as a single packet.

The following modes are supported:

- Port mode: Allows all traffic on a port to share a single virtual circuit across an MPLS network. Port mode uses virtual circuit type 5.
- VLAN mode: Transports Ethernet traffic from a source 802.1Q VLAN to a destination 802.1Q VLAN through a single virtual circuit over an MPLS network. VLAN mode uses virtual circuit type 5 as the default (does not transport dot1q tag); however, uses virtual circuit type 4 (transports dot1 tag) if the remote PE does not support virtual circuit type 5 for subinterface-based (VLAN-based) EoMPLS.

Interworking between EoMPLS port mode and EoMPLS VLAN mode: If EoMPLS port mode is configured on a local PE and EoMPLS VLAN mode on a remote PE, then the customer edge (CE) Layer 2 switchport interface must be configured as an *access* on the port mode side and the Spanning Tree Protocol must be disabled on the VLAN mode side of the CE device.

The maximum transmission unit (MTU) of all the intermediate links between PEs must be able to carry the largest Layer 2 packet received on ingress PE.

Starting with the Cisco IOS XE Bengaluru 17.6.1 release, you can forward Link Aggregation Control Protocol (LACP) and Port Aggregation Protocol (PAgP) packets over Ethernet-over-MPLS Pseudowire in the Port mode.

How to Configure Ethernet-over-MPLS

EoMPLS can be configured in the port mode or VLAN mode.

Configuring Ethernet-over-MPLS Port Mode

EoMPLS port mode can be configured using either the Xconnect mode or protocol CLI method.

Xconnect Mode

To configure EoMPLS port mode in Xconnect mode, perform the following task:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface TenGigabitEthernet1/0/36`	Defines the interface to be configured as a trunk, and enters interface configuration mode.
Step 4	**no switchport** **Example:** `Device(config-if)# no switchport`	Enters Layer 3 mode for physical ports only.
Step 5	**no ip address** **Example:** `Device(config-if)# no ip address`	Ensures that no IP address is assigned to the physical port.
Step 6	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Ensures that the device does not send keepalive messages.
Step 7	**xconnect** *peer-device-id vc-id* **encapsulation mpls** **Example:** `Device(config-if)# xconnect 10.1.1.1 962 encapsulation mpls`	Binds the attachment circuit to a pseudowire virtual circuit (VC). The syntax for this command is the same as for all other Layer 2 transports.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Protocol CLI Method

To configure EoMPLS port mode in protocol CLI mode, perform the following task:

Procedure

	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 dst-ip** **Example:** `Device(config)# port-channel load-balance dst-ip`	Sets the load distribution method to the destination IP address.
Step 4	**interface** *interface-id* **Example:** `Device(config)# interface TenGigabitEthernet1/0/21`	Defines the interface to be configured as a trunk, and enters interface configuration mode.
Step 5	**no switchport** **Example:** `Device(config-if)# no switchport`	Enters Layer 3 mode for physical ports only.
Step 6	**no ip address** **Example:** `Device(config-if)# no ip address`	Ensures that no IP address is assigned to the physical port.
Step 7	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Ensures that the device does not send keepalive messages.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 17`	Establishes a pseudowire interface with a value that you specify and enters pseudowire configuration mode.
Step 10	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 11	**neighbor** *peer-ip-addr vc-id* **Example:** `Device(config-if)# neighbor 10.10.0.10 17`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 12	**l2vpn xconnect context** *context-name* **Example:** `Device(config-if)# l2vpn xconnect context vpws17`	Creates an L2VPN cross connect context and enters Xconnect context configuration mode.
Step 13	**member** *interface-id* **Example:** `Device(config-if-xconn)# member TenGigabitEthernet1/0/21`	Specifies interface that forms an L2VPN cross connect.
Step 14	**member pseudowire** *number* **Example:** `Device(config-if-xconn)# member pseudowire 17`	Specifies the pseudowire interface that forms an L2VPN cross connect.

	Command or Action	Purpose
Step 15	**end** **Example:** `Device(config-if-xconn)# end`	Exits Xconnect interface configuration mode and returns to privileged EXEC mode.

Configuring Ethernet-over-MPLS VLAN Mode

EoMPLS VLAN mode can be configured using either the Xconnect mode or protocol-CLI method.

Xconnect Mode

To configure EoMPLS VLAN mode in Xconnect mode, perform the following task:

Procedure

	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** *interface-id* **Example:** `Device(config)# interface TenGigabitEthernet1/0/36`	Defines the interface to be configured as a trunk, and enters interface configuration mode.
Step 4	**no switchport** **Example:** `Device(config-if)# no switchport`	Enters Layer 3 mode, for physical ports only.
Step 5	**no ip address** **Example:** `Device(config-if)# no ip address`	Ensures that there is no IP address assigned to the physical port.

	Command or Action	Purpose
Step 6	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Ensures that the device does not send keepalive messages.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *interface-id.subinterface* **Example:** `Device(config)# interface TenGigabitEthernet1/0/36.1105`	Defines the subinterface to be configured, and enters subinterface configuration mode.
Step 9	**encapsulation dot1Q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1Q 1105`	Enables IEEE 802.1Q encapsulation of traffic on the subinterface.
Step 10	**xconnect** *peer-ip-addr vc-id* **encapsulation mpls** **Example:** `Device(config-subif)# xconnect 10.0.0.1 1105 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 11	**end** **Example:** `Device(config-subif-xconn)# end`	Returns to privileged EXEC mode.

Protocol CLI Method

To configure EoMPLS VLAN mode in protocol-CLI mode, perform the following task:

Procedure

	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 dst-ip** **Example:** `Device(config)# port-channel load-balance dst-ip`	Sets the load-distribution method to the destination IP address.
Step 4	**interface** *interface-id* **Example:** `Device(config)# interface TenGigabitEthernet1/0/36`	Defines the interface to be configured as a trunk, and enters interface configuration mode.
Step 5	**no switchport** **Example:** `Device(config-if)# no switchport`	Enters Layer 3 mode, for physical ports only.
Step 6	**no ip address** **Example:** `Device(config-if)# no ip address`	Ensures that there is no IP address assigned to the physical port.
Step 7	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Ensures that the device does not send keepalive messages.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**interface** *interface-id.subinterface* **Example:** `Device(config)# interface TenGigabitEthernet1/0/36.1105`	Defines the subinterface to be configured, and enters subinterface configuration mode.
Step 10	**encapsulation dot1Q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1Q 1105`	Enables IEEE 802.1Q encapsulation of traffic on the subinterface.
Step 11	**exit** **Example:** `Device(config-subif)# exit`	Exits subinterface configuration mode and returns to interface configuration mode.
Step 12	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 17`	Establishes a pseudowire interface with a value that you specify and enters pseudowire configuration mode.
Step 13	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 14	**neighbor** *peer-ip-addr vc-id* **Example:** `Device(config-if)# neighbor 10.10.0.10 17`	Specifies the peer IP address and VC ID value of a L2VPN pseudowire.

	Command or Action	Purpose
Step 15	**l2vpn xconnect context** *context-name* **Example:** `Device(config-if)# l2vpn xconnect context vpws17`	Creates a L2VPN cross connect context, and enters Xconnect context configuration mode.
Step 16	**member** *interface-id.subinterface* **Example:** `Device(config-if-xconn)# member TenGigabitEthernet1/0/36.1105`	Specifies the subinterface that forms a L2VPN cross connect.
Step 17	**member pseudowire** *number* **Example:** `Device(config-if-xconn)# member pseudowire 17`	Specifies pseudowire interface that forms a L2VPN cross connect.
Step 18	**end** **Example:** `Device(config-if-xconn)# end`	Exits Xconnect configuration mode and returns to privileged EXEC mode.

Configuration Examples for Ethernet-over-MPLS

Figure 1: EoMPLS Topology

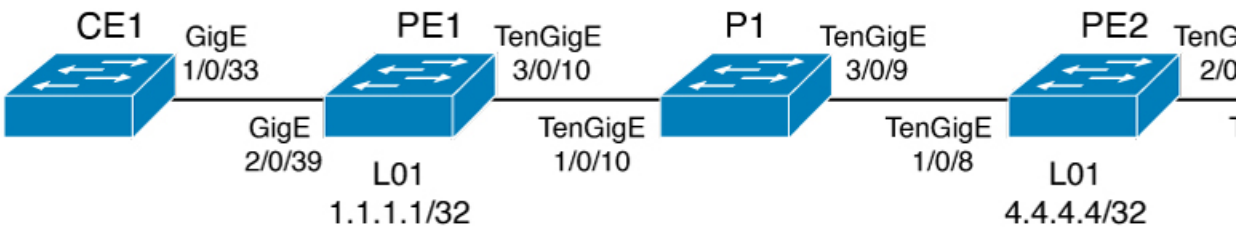

Table 1: EoMPLS Port Mode Configuration

PE Configuration	CE Configuration
mpls ip mpls label protocol ldp mpls ldp graceful-restart mpls ldp router-id loopback 1 force interface Loopback1 ip address 10.1.1.1 255.255.255.255 ip ospf 100 area 0 router ospf 100 router-id 10.1.1.1 nsf system mtu 9198 port-channel load-balance dst-ip ! interface gigabitethernet 2/0/39 no switchport no ip address no keepalive ! interface pseudowire101 encapsulation mpls neighbor 10.10.10.10 101 load-balance flow ip dst-ip load-balance flow-label both l2vpn xconnect context pw101 member pseudowire101 member gigabitethernet 2/0/39 ! interface tengigabitethernet 3/0/10 switchport trunk allowed vlan 142 switchport mode trunk channel-group 42 mode active ! interface Port-channel42 switchport trunk allowed vlan 142 switchport mode trunk ! interface Vlan142 ip address 10.11.11.11 255.255.255.0 ip ospf 100 area 0 mpls ip mpls label protocol ldp !	interface gigabitethernet 1/0/33 switchport trunk allowed vlan 912 switchport mode trunk spanning-tree portfast trunk ! interface Vlan912 ip address 10.91.2.3 255.255.255.0 !

Table 2: EoMPLS VLAN Mode Configuration

PE Configuration	CE Configuration
interface tengigabitethernet 1/0/36 no switchport no ip address no keepalive exit ! interface tengigabitethernet 1/0/36.1105 encapsulation dot1Q 1105 exit ! interface pseudowire1105 encapsulation mpls neighbor 10.10.0.10 1105 exit ! l2vpn xconnect context vme1105 member tengigabitethernet 1/0/36.1105 member pseudowire1105 end !	interface fortygigabitethernet 1/9 switchport switchport mode trunk switchport trunk allowed vlan 1105 mtu 9216 end !

Table 3: Interworking Between EoMPLS Port Mode and EoMPLS VLAN Mode Configuration

PE Configuration: Port Mode	CE Configuration: Port Mode
interface tengigabitethernet 1/0/37 no switchport no ip address no keepalive exit ! interface pseudowire1105 encapsulation mpls neighbor 10.11.11.11 1105 exit ! l2vpn xconnect context vme1105 member tengigabitethernet 1/0/37 member pseudowire1105 end !	interface fortygigabitethernet1/10 switchport switchport mode access switchport access vlan 1105 end no spanning-tree vlan 1105 !

PE Configuration: VLAN Mode	CE Configuration: VLAN Mode
interface tengigabitethernet 1/0/36 no switchport no ip address no keepalive exit ! interface tengigabitethernet 1/0/36.1105 encapsulation dot1Q 1105 exit ! interface pseudowire1105 encapsulation mpls neighbor 10.10.0.10 1105 exit ! l2vpn xconnect context vme1105 member tengigabitethernet 1/0/36.1105 member pseudowire1105 end !	interface fortygigabitethernet 1/9 switchport switchport mode trunk switchport trunk allowed vlan 1105 mtu 9216 end no spanning-tree vlan 1105 !

Another scenario for interworking between EoMPLS port mode and EoMPLS VLAN mode is to configure the following commands on both CE devices:

- **switchport mode trunk**
- **switchport trunk allowed vlan** *vlan-id*
- **spanning-tree vlan** *vlan-id*

Data traffic will flow through by disabling STP on both CE devices, if the traffic sent is not double VLAN tagged.

The following is a sample output of the **show mpls l2 vc vcid** *vc-id* **detail** command:

```
Device# show mpls l2 vc vcid 1105 detail
Local interface: TenGigabitEthernet1/0/36.1105 up, line protocol up, Eth VLAN 1105 up
  Interworking type is Ethernet
  Destination address: 10.0.0.1, VC ID: 1105, VC status: up
    Output interface: Po10, imposed label stack {33 10041}
    Preferred path: not configured
    Default path: active
    Next hop: 10.10.0.1
  Create time: 00:04:09, last status change time: 00:02:13
    Last label FSM state change time: 00:02:12
  Signaling protocol: LDP, peer 10.0.0.1:0 up
    Targeted Hello: 10.0.0.10(LDP Id) -> 10.0.0.1, LDP is UP
    Graceful restart: configured and enabled
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
     Last local PW i/f circ status rcvd: No fault
```

```
    Last local LDP TLV     status sent: No fault
    Last remote LDP TLV    status rcvd: No fault
    Last remote LDP ADJ    status rcvd: No fault
  MPLS VC labels: local 124, remote 10041
  Group ID: local 336, remote 352
  MTU: local 9198, remote 9198
  Remote interface description:
  MAC Withdraw: sent:1, received:0
Sequencing: receive disabled, send disabled
Control Word: On (configured: autosense)
SSO Descriptor: 10.0.0.1/1105, local label: 124
Dataplane:
  SSM segment/switch IDs: 9465983/446574 (used), PWID: 109
VC statistics:
  transit packet totals: receive 0, send 0
  transit byte totals:   receive 0, send 0
  transit packet drops:  receive 0, seq error 0, send 0
```

The following is a sample output of the **show l2vpn atom vc vcid** *vc-id* **detail** command:

```
Device# show l2vpn atom vc vcid 1105 detail
pseudowire100109 is up, VC status is up PW type: Ethernet
  Create time: 00:04:17, last status change time: 00:02:22
    Last label FSM state change time: 00:02:20
  Destination address: 10.0.0.1 VC ID: 1105
    Output interface: Po10, imposed label stack {33 10041}
    Preferred path: not configured
    Default path: active
    Next hop: 10.10.0.1
  Member of xconnect service TenGigabitEthernet1/0/36.1105-1105, group right
    Associated member TenGigabitEthernet1/0/36.1105 is up, status is up
    Interworking type is Ethernet
    Service id: 0x1f000037
  Signaling protocol: LDP, peer 10.0.0.1:0 up
    Targeted Hello: 10.0.0.10(LDP Id) -> 10.0.0.1, LDP is UP
    Graceful restart: configured and enabled
    Non stop routing: not configured and not enabled
    PWid FEC (128), VC ID: 1105
    Status TLV support (local/remote)         : enabled/supported
      LDP route watch                         : enabled
      Label/status state machine              : established, LruRru
      Local dataplane status received         : No fault
      BFD dataplane status received           : Not sent
      BFD peer monitor status received        : No fault
      Status received from access circuit     : No fault
      Status sent to access circuit           : No fault
      Status received from pseudowire i/f     : No fault
      Status sent to network peer             : No fault
      Status received from network peer       : No fault
      Adjacency status of remote peer         : No fault
  Sequencing: receive disabled, send disabled
  Bindings
    Parameter    Local                          Remote
    ------------ ------------------------------ ------------------------------
    Label        124                            10041
    Group ID     336                            352
    Interface
    MTU          9198                           9198
    Control word on (configured: autosense)     on
    PW type      Ethernet                       Ethernet
    VCCV CV type 0x02                           0x02
                   LSPV [2]                       LSPV [2]
    VCCV CC type 0x06                           0x06
                   RA [2], TTL [3]                RA [2], TTL [3]
```

```
  Status TLV   enabled                          supported
SSO Descriptor: 10.0.0.1/1105, local label: 124
Dataplane:
  SSM segment/switch IDs: 9465983/446574 (used), PWID: 109
Rx Counters
  0 input transit packets, 0 bytes
  0 drops, 0 seq err
  0 MAC withdraw
Tx Counters
  0 output transit packets, 0 bytes
  0 drops
  1 MAC withdraw
```

The following is a sample output of the **show mpls forwarding-table** command:

```
Device# show mpls forwarding-table 10.0.0.1

Local      Outgoing   Prefix           Bytes Label   Outgoing       Next Hop
Label      Label      or Tunnel Id     Switched      interface
2049       33         10.0.0.1/32      38540         Hu2/0/30/2.1   10.0.0.2
           33         10.0.0.1/32      112236        Hu2/0/30/2.2   10.0.0.6
           33         10.0.0.1/32      46188         Hu2/0/30/2.3   10.0.0.8
```

Configuring Pseudowire Redundancy

This section provides information about how to configure pseudowire redundancy.

Prerequisites for Pseudowire Redundancy

- Configure the **no switchport**, **no keepalive**, and **no ip address** before configuring Xconnect mode to connect the attachment circuit.
- For load-balancing, configure the **port-channel load-balance** command.
- Subinterfaces must be supported to enable pseudowire redundancy VLAN mode.

Restrictions for Pseudowire Redundancy

The following sections list the restrictions for pseudowire redundancy port mode and pseudowire redundancy VLAN mode.

Restrictions for Pseudowire Redundancy Port Mode

- Ethernet Flow Point (EFP) and Internet Group Management Protocol (IGMP) Snooping is not supported.
- Flow Label for ECMP load balancing in a core network is based on customer's source IP, destination IP, source MAC and destination MAC.
- MPLS QoS is supported in Pipe and Uniform Mode. Default mode is Pipe Mode.
- QoS: Customer DSCP Re-marking is not supported with VPWS and EoMPLS.
- VCCV Ping with explicit null is not supported.

- The **ip unnumbered** command is not supported in MPLS configuration.
- Not more than one backup pseudowire is supported.
- PW redundancy group switchover is not supported

Restrictions for Pseudowire Redundancy VLAN Mode

- Virtual circuit will not work if the same interworking type is not configured on PE devices.
- Untagged traffic is not supported as incoming traffic.
- Xconnect mode cannot be enabled on Layer 2 subinterfaces because multiplexer user-network interface (MUX UNI) is not supported.
- Xconnect mode cannot be configured on subinterfaces if it is enabled on the main interface for port-to-port transport.
- Flow Aware Transport (FAT) can be configured on Protocol CLI mode only.
- MACsec is not supported on pseudowire redundancy VLAN mode.
- QoS: Customer DSCP Remarking is not supported with VPWS and pseudowire redundancy.
- MPLS QoS is supported only in pipe and uniform mode. Default mode is pipe mode.
- In VLAN mode pseudowire redundancy, Cisco Discovery Protocol packets from the CE will be processed by the PE, but is not carried over the pseudowire redundancy virtual circuit, whereas in port mode, Cisco Discovery Protocol packets from the CE will be carried over the virtual circuit.
- Only Ethernet and VLAN interworking types are supported.
- L2 Protocol Tunneling is not supported.

Information About Pseudowire Redundancy

The L2VPN pseudowire redundancy feature enables you to configure your network to detect a failure in the network and reroute the Layer 2 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) device or of the link between the PE and customer edge (CE) devices.

The maximum transmission unit (MTU) of all the intermediate links between PEs must be able to carry the largest Layer 2 packet received on ingress PE.

Pseudowire redundancy can be configured using both the Xconnect and the protocol CLI method.

How to Configure Pseudowire Redundancy

Pseudowire redundancy can be configured in the port mode or VLAN mode.

Configuring Pseudowire Redundancy Port Mode

Pseudowire redundancy port-mode can be configured using either the Xconnect mode or protocol-CLI method.

Xconnect Mode

To configure pseudowire redundancy port mode in Xconnect mode, perform the following task:

Note To enable load balance, use the corresponding **load-balance** commands from Xconnect Mode procedure of the 'How to Configure Ethernet-over-MPLS section.

Procedure

	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** *interface-id* **Example:** `Device(config)# interface GigabitEthernet1/0/44`	Defines the interface to be configured as a trunk, and enters interface configuration mode.
Step 4	**no switchport** **Example:** `Device(config-if)# no switchport`	Enters Layer 3 mode, for physical ports only.
Step 5	**no ip address** **Example:** `Device(config-if)# no ip address`	Ensures that there is no IP address assigned to the physical port.
Step 6	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Ensures that the device does not send keepalive messages.

	Command or Action	Purpose
Step 7	**xconnect** *peer-device-id vc-id* **encapsulation mpls** **Example:** `Device(config-if)# xconnect 10.1.1.1 117 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 8	**backup peer** *peer-router-ip-addr* **vcid** *vc-id* [**priority** *value*] **Example:** `Device(config-if)# backup peer 10.11.11.11 118 priority 9`	Specifies a redundant peer for a pseudowire VC.
Step 9	**end** **Example:** `Device(config)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Protocol CLI Method

To configure pseudowire redundancy port mode in protocol CLI mode, perform the following task:

Procedure

	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 dst-ip** **Example:** `Device(config)# port-channel`	Sets the load-distribution method to the destination IP address.

	Command or Action	Purpose
	`load-balance dst-ip`	
Step 4	**interface** *interface-id* **Example:** `Device(config)# interface TenGigabitEthernet1/0/36`	Defines the interface to be configured as a trunk, and enters interface configuration mode.
Step 5	**no switchport** **Example:** `Device(config-if)# no switchport`	Enters Layer 3 mode, for physical ports only.
Step 6	**no ip address** **Example:** `Device(config-if)# no ip address`	Ensures that there is no IP address assigned to the physical port.
Step 7	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Ensures that the device does not send keepalive messages.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 9	**interface pseudowire** *number-active* **Example:** `Device(config)# interface pseudowire 17`	Establishes an active pseudowire interface with a value that you specify and enters pseudowire configuration mode.
Step 10	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies the tunneling encapsulation.

	Command or Action	Purpose
Step 11	**neighbor** *active-peer-ip-addr vc-id* **Example:** `Device(config-if)# neighbor 10.10.0.10 17`	Specifies the active peer IP address and VC ID value of a L2VPN pseudowire.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 13	**interface pseudowire** *number-standby* **Example:** `Device(config)# interface pseudowire 18`	Establishes a standby pseudowire interface with a value that you specify and enters pseudowire configuration mode.
Step 14	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 15	**neighbor** *standby-peer-ip-addr vc-id* **Example:** `Device(config-if)# neighbor 10.10.0.11 18`	Specifies the standby peer IP address and VC ID value of a L2VPN pseudowire.
Step 16	**l2vpn xconnect context** *context-name* **Example:** `Device(config-if)# l2vpn xconnect context vpws17`	Creates a L2VPN cross connect context, and attaches the VLAN mode EoMPLS attachment circuit to the active and standby pseudowire interfaces.
Step 17	**member** *interface-id* **Example:** `Device(config-if-xconn)# member TenGigabitEthernet1/0/36`	Specifies interface that forms a L2VPN cross connect.

	Command or Action	Purpose
Step 18	**member pseudowire** *number-active* **group** *group-name* [**priority** *value*] **Example:** `Device(config-if-xconn)# member pseudowire 17 group pwr10`	Specifies active pseudowire interface that forms a L2VPN cross connect.
Step 19	**member pseudowire** *number-standby* **group** *group-name* [**priority** *value*] **Example:** `Device(config-if-xconn)# member pseudowire 18 group pwr10 priority 6`	Specifies standby pseudowire interface that forms a L2VPN cross connect.
Step 20	**end** **Example:** `Device(config-if-xconn)# end`	Exits Xconnect configuration mode and returns to privileged EXEC mode.

Configuring Pseudowire Redundancy VLAN Mode

Pseudowire redundancy VLAN mode can be configured using either the Xconnect mode or the protocol CLI method.

Xconnect Mode

To configure pseudowire redundancy VLAN mode in Xconnect mode, perform the following task:

Procedure

	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	**interface** *interface-id* **Example:** `Device(config)# interface TenGigabitEthernet1/0/36`	Defines the interface to be configured as a trunk, and enters interface configuration mode.
Step 4	**no switchport** **Example:** `Device(config-if)# no switchport`	Enters Layer 3 mode for physical ports only.
Step 5	**no ip address** **Example:** `Device(config-if)# no ip address`	Ensures that no IP address is assigned to the physical port.
Step 6	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Ensures that the device does not send keepalive messages.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *interface-id.subinterface* **Example:** `Device(config)# interface TenGigabitEthernet1/0/36.1105`	Defines the subinterface to be configured, and enters subinterface configuration mode.
Step 9	**encapsulation dot1Q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1Q 1105`	Enables IEEE 802.1Q encapsulation of traffic on the subinterface.

	Command or Action	Purpose
Step 10	**xconnect** *peer-ip-addr vc-id* **encapsulation mpls** **Example:** `Device(config-subif)# xconnect 10.0.0.1 1105 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 11	**backup peer** *peer-ip-addr vc-id* **[priority** *value*] **Example:** `Device(config-subif-xconn)# backup peer 10.10.10.10 1105 priority 8`	Specifies a redundant peer for the pseudowire VC.
Step 12	**end** **Example:** `Device(config-subif-xconn)# end`	Exits Xconnect configuration mode and returns to privileged EXEC mode.

Protocol CLI Method

To configure pseudowire redundancy VLAN mode in protocol CLI mode, perform the following task:

Procedure

	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 dst-ip** **Example:** `Device(config)# port-channel`	Sets the load-distribution method to the destination IP address.

	Command or Action	Purpose
	`load-balance dst-ip`	
Step 4	**interface** *interface-id* **Example:** `Device(config)# interface TenGigabitEthernet1/0/36`	Defines the interface to be configured as a trunk, and enters interface configuration mode.
Step 5	**no switchport** **Example:** `Device(config-if)# no switchport`	Enters Layer 3 mode for physical ports only.
Step 6	**no ip address** **Example:** `Device(config-if)# no ip address`	Ensures that there is no IP address assigned to the physical port.
Step 7	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Ensures that the device does not send keepalive messages.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 9	**interface** *interface-id.subinterface* **Example:** `Device(config)# interface TenGigabitEthernet1/0/36.1105`	Defines the subinterface to be configured, and enters subinterface configuration mode.
Step 10	**encapsulation dot1Q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1Q 1105`	Enables IEEE 802.1Q encapsulation of traffic on the subinterface.

	Command or Action	Purpose
Step 11	**exit** **Example:** `Device(config-subif)# exit`	Exits subinterface configuration mode.
Step 12	**interface pseudowire** *number-active* **Example:** `Device(config)# interface pseudowire 17`	Establishes an active pseudowire interface with a value that you specify, and enters pseudowire configuration mode.
Step 13	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 14	**neighbor** *active-peer-ip-addr vc-id* **Example:** `Device(config-if)# neighbor 10.10.0.10 17`	Specifies the active peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 15	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 16	**interface pseudowire** *number-standby* **Example:** `Device(config)# interface pseudowire 18`	Establishes a standby pseudowire interface with a value that you specify, and enters pseudowire configuration mode.
Step 17	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies the tunneling encapsulation.

	Command or Action	Purpose
Step 18	**neighbor** *standby-peer-ip-addr vc-id* **Example:** `Device(config-if)# neighbor 10.10.0.11 18`	Specifies the standby peer IP address and VC ID value of an L2VPN pseudowire.
Step 19	**l2vpn xconnect context** *context-name* **Example:** `Device(config-if)# l2vpn xconnect context vpws17`	Creates an L2VPN cross-connect context, and attaches the VLAN mode EoMPLS attachment circuit to the active and standby pseudowire interfaces.
Step 20	**member** *interface-id.subinterface* **Example:** `Device(config-if-xconn)# member TenGigabitEthernet1/0/36.1105`	Specifies the interface that forms an L2VPN cross connect.
Step 21	**member pseudowire** *number-active* **group** *group-name* [**priority** *value*] **Example:** `Device(config-if-xconn)# member pseudowire 17 group pwr10`	Specifies the active pseudowire interface that forms an L2VPN cross connect.
Step 22	**member pseudowire** *number-standby* **group** *group-name* [**priority** *value*] **Example:** `Device(config-if-xconn)# member pseudowire 18 group pwr10 priority 6`	Specifies standby pseudowire interface that forms an L2VPN cross connect.
Step 23	**end** **Example:** `Device(config-if-xconn)# end`	Exits Xconnect configuration mode and returns to privileged EXEC mode.

Configuration Examples for Pseudowire Redundancy

Table 4: Pseudowire Redundancy Port Mode Configuration

PE Configuration	CE Configuration
mpls ip mpls label protocol ldp mpls ldp graceful-restart mpls ldp router-id loopback 1 force ! interface Loopback1 ip address 10.1.1.1 255.255.255.255 ip ospf 100 area 0 router ospf 100 router-id 10.1.1.1 nsf ! interface gigabitethernet 2/0/39 no switchport no ip address no keepalive ! interface pseudowire101 encapsulation mpls neighbor 10.10.10.10 101 ! interface pseudowire102 encapsulation mpls neighbor 10.10.10.11 101 l2vpn xconnect context pw101 member pseudowire101 group pwgrp1 priority 1 member pseudowire102 group pwgrp1 priority 15 member GigabitEthernet2/0/39 ! interface tengigabitethernet 3/0/10 switchport trunk allowed vlan 142 switchport mode trunk channel-group 42 mode active ! interface Port-channel42 switchport trunk allowed vlan 142 switchport mode trunk ! interface Vlan142 ip address 10.11.11.11 255.255.255.0 ip ospf 100 area 0 mpls ip mpls label protocol ldp !	interface gigabitethernet 1/0/33 switchport trunk allowed vlan 912 switchport mode trunk spanning-tree portfast trunk ! interface Vlan912 ip address 10.91.2.3 255.255.255.0 !

Table 5: Pseudowire Redundancy VLAN Mode Configuration

PE Configuration	CE Configuration
interface tengigabitethernet 1/0/36 no switchport no ip address no keepalive exit ! interface tengigabitethernet 1/0/36.1105 encapsulation dot1Q 1105 exit ! interface pseudowire1105 encapsulation mpls neighbor 10.10.0.10 1105 exit ! interface pseudowire1106 encapsulation mpls neighbor 10.10.0.11 1106 ! l2vpn xconnect context vme1105 member tengigabitethernet 1/0/36.1105 member pseudowire1105 group pwr10 member pseudowire1106 group pwr10 priority 6 end !	interface fortygigabitethernet 1/9 switchport switchport mode trunk switchport trunk allowed vlan 1105 mtu 9216 end !

The following is a sample output of the **show mpls l2 vc vcid** *vc-id* **detail** command:

```
Device# show mpls l2 vc vcid 1105 detail
Local interface: TenGigabitEthernet1/0/36.1105 up, line protocol up, Eth VLAN 1105 up
  Interworking type is Ethernet
  Destination address: 10.11.11.11, VC ID: 1105, VC status: standby
    Output interface: Po10, imposed label stack {1616}
    Preferred path: not configured
    Default path: active
    Next hop: 10.10.0.1
  Create time: 00:04:09, last status change time: 00:02:13
    Last label FSM state change time: 00:02:15
  Signaling protocol: LDP, peer 10.11.11.11:0 up
    Targeted Hello: 10.10.0.10(LDP Id) -> 10.11.11.11, LDP is UP
    Graceful restart: configured and enabled
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LrdRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: DOWN(standby)
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: DOWN(standby)
      Last remote LDP TLV    status rcvd: No fault
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 125, remote 1616
    Group ID: local 336, remote 0
```

```
    MTU: local 9198, remote 9198
    Remote interface description:
    MAC Withdraw: sent:1, received:0
  Sequencing: receive disabled, send disabled
  Control Word: On (configured: autosense)
  SSO Descriptor: 10.11.11.11/1105, local label: 125
  Dataplane:
    SSM segment/switch IDs: 96143/450671 (used), PWID: 110
  VC statistics:
    transit packet totals: receive 0, send 0
    transit byte totals:   receive 0, send 0
    transit packet drops:  receive 0, seq error 0, send 0
```

The following is a sample output of the **show l2vpn atom vc vcid** *vc-id* **detail** command:

```
Device# show l2vpn atom vc vcid 1105 detail
pseudowire100110 is up, VC status is standby PW type: Ethernet
  Create time: 00:04:17, last status change time: 00:02:22
    Last label FSM state change time: 00:02:24
  Destination address: 10.11.11.11 VC ID: 1105
    Output interface: Po10, imposed label stack {1616}
    Preferred path: not configured
    Default path: active
    Next hop: 10.0.0.1
  Member of xconnect service TenGigabitEthernet1/0/36.1105-1105, group right
    Associated member TenGigabitEthernet1/0/36.1105 is up, status is up
    Interworking type is Ethernet
    Service id: 0x1f000037
  Signaling protocol: LDP, peer 10.11.11.11:0 up
    Targeted Hello: 10.0.0.10(LDP Id) -> 10.11.11.11, LDP is UP
    Graceful restart: configured and enabled
    Non stop routing: not configured and not enabled
    PWid FEC (128), VC ID: 1105
    Status TLV support (local/remote)         : enabled/supported
      LDP route watch                         : enabled
      Label/status state machine              : established, LrdRru
      Local dataplane status received         : No fault
      BFD dataplane status received           : Not sent
      BFD peer monitor status received        : No fault
      Status received from access circuit     : DOWN(standby)
      Status sent to access circuit           : No fault
      Status received from pseudowire i/f     : No fault
      Status sent to network peer             : DOWN(standby)
      Status received from network peer       : No fault
      Adjacency status of remote peer         : No fault
  Sequencing: receive disabled, send disabled
  Bindings
    Parameter    Local                          Remote
    ------------ ------------------------------ ------------------------------
    Label        125                            1616
   Group ID      336                            0
    Interface
    MTU          9198                           9198
    Control word on (configured: autosense)     on
    PW type      Ethernet                       Ethernet
    VCCV CV type 0x02                           0x02
                   LSPV [2]                       LSPV [2]
    VCCV CC type 0x06                           0x02
                   RA [2], TTL [3]                RA [2]
    Status TLV   enabled                        supported
  SSO Descriptor: 10.11.11.11/1105, local label: 125
  Dataplane:
    SSM segment/switch IDs: 96143/450671 (used), PWID: 110
  Rx Counters
```

```
    0 input transit packets, 0 bytes
    0 drops, 0 seq err
    0 MAC withdraw
  Tx Counters
    0 output transit packets, 0 bytes
    0 drops
    1 MAC withdraw
```

The following is a sample output of the **show mpls l2transport vc** *vc-id* command:

```
Device# show mpls l2transport vc 101

Local intf                     Local circuit     Dest address   VC ID     Status
-----------------------------  ---------------   ------------   --------  ----------
TenGigabitEthernet1/0/36.1105  Eth VLAN 1105     10.0.0.1       1105      UP
TenGigabitEthernet1/0/36.1105  Eth VLAN 1105     10.11.11.11    1105      STANDBY
```

Feature History for Ethernet-over-MPLS and Pseudowire Redundancy

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	Ethernet-over-MPLS and Pseudowire Redundancy	Ethernet-over-MPLS is one of the Any Transport over MPLS (AToM) transport types. EoMPLS works by encapsulating Ethernet protocol data units (PDUs) in MPLS packets and forwarding them across the MPLS network. Each PDU is transported as a single packet. The L2VPN pseudowire redundancy feature enables you to configure your network to detect a failure in the network and reroute the Layer 2 service to another endpoint that can continue to provide service. Port mode support is introduced. Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.1	Ethernet-over-MPLS and Pseudowire Redundancy	Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.
Cisco IOS XE Gibraltar 16.12.1	VLAN support for Ethernet-over-MPLS	EoMPLS VLAN mode can be configured using either the Xconnect mode or protocol-CLI method.
Cisco IOS XE Amsterdam 17.1.1	Macsec over EoMPLS	In VLAN mode EoMPLS, only those packets configured with macsec dot1q-in-clear 1 command on the CE device will be processed by the PE device.
Cisco IOS XE Bengaluru 17.6.1	LACP and PAgP packet forwarding over Ethernet-over-MPLS Pseudowire	Support for forwarding Link Aggregation Control Protocol (LACP) and Port Aggregation Protocol (PAgP) packets over Ethernet-over-MPLS Pseudowire in the Port mode was introduced.
Cisco IOS XE Cupertino 17.7.1	Ethernet-over-MPLS	This feature was implemented on the C9500X-28C8D and C9500X-60L4D models.
Cisco IOS XE Dublin 17.11.1	Pseudowire Redundancy	This feature was implemented on the C9500X-28C8D and C9500X-60L4D models.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/

http://www.cisco.com/go/cfn.