



Configuring Ethernet-over-MPLS (EoMPLS)

- [Prerequisites for Ethernet-over-MPLS, on page 1](#)
- [Restrictions for Ethernet-over-MPLS, on page 1](#)
- [Information About Ethernet-over-MPLS, on page 3](#)
- [How to Configure Ethernet-over-MPLS, on page 3](#)
- [Configuration Examples for Ethernet-over-MPLS, on page 11](#)
- [Feature Information for Ethernet-over-MPLS \(EoMPLS\), on page 16](#)

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 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.
- Link Aggregation Control Protocol (LACP) and Port Aggregation Protocol (PAgP) packets are not forwarded over Ethernet-over-MPLS Pseudowire, as these are processed by the local PE.

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 CLI is not supported.

- Link Aggregation Control Protocol (LACP) and Port Aggregation Protocol (PAgP) packets are not forwarded over Ethernet-over-MPLS Pseudowire, as these are processed by the local PE.

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.

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. |

| | Command or Action | Purpose |
|---------------|--|---|
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface <i>interface-id</i> 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 <i>peer-device-id</i> <i>vc-id</i> 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: | Ensures that the device does not send keepalive messages. |

| | Command or Action | Purpose |
|----------------|---|--|
| | Device (config-if) # no keepalive | |
| Step 8 | exit Example: Device (config-if) # exit | Exits interface configuration mode and returns to global configuration mode. |
| Step 9 | interface pseudowire <i>number</i> 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 <i>peer-ip-addr vc-id</i> 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 <i>context-name</i> Example: Device (config-if) # l2vpn xconnect context vpws17 | Creates an L2VPN cross connect context and enters Xconnect context configuration mode. |
| Step 13 | member <i>interface-id</i> Example: Device (config-if-xconn) # member TenGigabitEthernet1/0/21 | Specifies interface that forms an L2VPN cross connect. |

| | Command or Action | Purpose |
|----------------|---|--|
| Step 14 | member pseudowire <i>number</i> Example: <pre>Device(config-if-xconn)# member pseudowire 17</pre> | Specifies the pseudowire interface that forms an L2VPN cross connect. |
| Step 15 | end Example: <pre>Device(config-if-xconn)# end</pre> | 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: <pre>Device> enable</pre> | Enables privileged EXEC mode. Enter your password if prompted. |
| Step 2 | configure terminal Example: <pre>Device# configure terminal</pre> | Enters global configuration mode. |
| Step 3 | interface <i>interface-id</i> Example: <pre>Device(config)# interface TenGigabitEthernet1/0/36</pre> | Defines the interface to be configured as a trunk, and enters interface configuration mode. |
| Step 4 | no switchport Example: <pre>Device(config-if)# no switchport</pre> | Enters Layer 3 mode, for physical ports only. |

| | Command or Action | Purpose |
|----------------|--|---|
| 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. |
| 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 <i>interface-id.subinterface</i> Example: Device (config) # interface TenGigabitEthernet1/0/36.1105 | Defines the subinterface to be configured, and enters subinterface configuration mode. |
| Step 10 | encapsulation dot1Q <i>vlan-id</i> 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 <i>number</i> 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 <i>peer-ip-addr vc-id</i> 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 <i>context-name</i> Example: <pre>Device(config-if)# l2vpn xconnect context vpws17</pre> | Creates a L2VPN cross connect context, and enters Xconnect context configuration mode. |
| Step 16 | member <i>interface-id.subinterface</i> Example: <pre>Device(config-if-xconn)# member TenGigabitEthernet1/0/36.1105</pre> | Specifies the subinterface that forms a L2VPN cross connect. |
| Step 17 | member pseudowire <i>number</i> Example: <pre>Device(config-if-xconn)# member pseudowire 17</pre> | Specifies pseudowire interface that forms a L2VPN cross connect. |
| Step 18 | end Example: <pre>Device(config-if-xconn)# end</pre> | Exits Xconnect configuration mode and returns to privileged EXEC mode. |

Configuration Examples for Ethernet-over-MPLS

Figure 1: EoMPLS Topology

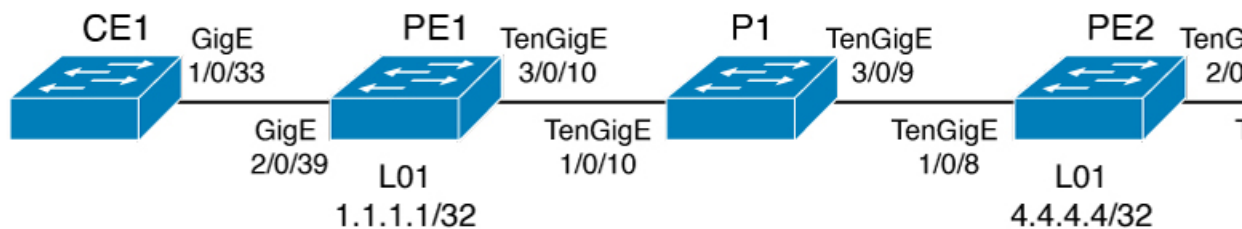


Table 1: EoMPLS Port Mode Configuration

| PE Configuration | CE Configuration |
|---|---|
| <pre> 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 ! </pre> | <pre> 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 ! </pre> |

Table 2: EoMPLS VLAN Mode Configuration

| PE Configuration | CE Configuration |
|---|---|
| <pre> 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 ! </pre> | <pre> interface fortygigabitethernet 1/9 switchport switchport mode trunk switchport trunk allowed vlan 1105 mtu 9216 end ! </pre> |

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

| PE Configuration: Port Mode | CE Configuration: Port Mode |
|--|--|
| <pre> 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 ! </pre> | <pre> interface fortygigabitethernet1/10 switchport switchport mode access switchport access vlan 1105 end no spanning-tree vlan 1105 ! </pre> |

| PE Configuration: VLAN Mode | CE Configuration: VLAN Mode |
|---|---|
| <pre>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 !</pre> | <pre>interface fortygigabitethernet 1/9 switchport switchport mode trunk switchport trunk allowed vlan 1105 mtu 9216 end no spanning-tree vlan 1105 !</pre> |

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 Label | Outgoing Label | Prefix or Tunnel Id | Bytes Switched | Outgoing interface | Next Hop |
|-------------|----------------|---------------------|----------------|--------------------|----------|
| 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 |

Feature Information for Ethernet-over-MPLS (EoMPLS)

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

These features are available on all 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. The Layer 2 VPN 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. |
| Cisco IOS XE Gibraltar 16.12.1 | VLAN mode support for Ethernet-over-MPLS | 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. |
| Cisco IOS XE Amsterdam 17.1.1 | Macsec over EoMPLS | In VLAN mode, the switch (PE device) can now process packets in which the 802.1Q tag is not encrypted by the CE device. |

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.

