



## Configure Point-to-Point Layer 2 Services

Point-to-point service basically emulates a transport circuit between two end nodes so the end nodes appear to be directly connected over a point-to-point link. This can be used to connect two sites.

This section introduces you to point-to-point Layer 2 services, and also describes the configuration procedures to implement it.

The following point-to-point services are supported:

- **Local Switching**—A point-to-point internal circuit on a router, also known as local connect. Local switching allows switching of Layer 2 data between two attachment circuits on the same device.
- **Attachment circuit**—An attachment circuit (AC) is a physical or logical port or circuit that connects a CE device to a PE device.
- **Pseudowires**—A virtual point-to-point circuit from one PE router to another. Pseudowires are implemented over the MPLS network.



**Note** Point-to-point Layer 2 services are also called as MPLS Layer 2 VPNs.

- [Pseudowire over MPLS](#) , on page 1
- [PW over MPLS Supported Modes](#), on page 5
- [Virtual Circuit Connection Verification on L2VPN](#), on page 15
- [Preferred tunnel path](#), on page 16
- [Configure Local Switching Between Attachment Circuits](#), on page 23
- [MPLS PW Traffic Load Balancing on P Router](#), on page 25
- [L2VPN Traffic Load Balancing on PE Router](#), on page 29

## Pseudowire over MPLS

*Table 1: Feature History Table*

Feature Name	Release Information	Feature Description

Pseudowire over MPLS	Release 7.3.15	This feature allows you to tunnel two L2VPN Provider Edge (PE) devices to transport L2VPN traffic over an MPLS core network. MPLS labels are used to transport data over the pseudowire.
----------------------	----------------	--

A pseudowire (PW) is a point-to-point connection between two provider edge (PE) devices which connects two attachment circuits (ACs). The two ACs connected at each PE are linked by a PW over the MPLS network, which is the MPLS PW.

PWs provide a common intermediate format to transport multiple types of network services over a Packet Switched Network (PSN) – a network that forwards packets – IPv4, IPv6, MPLS, Ethernet.

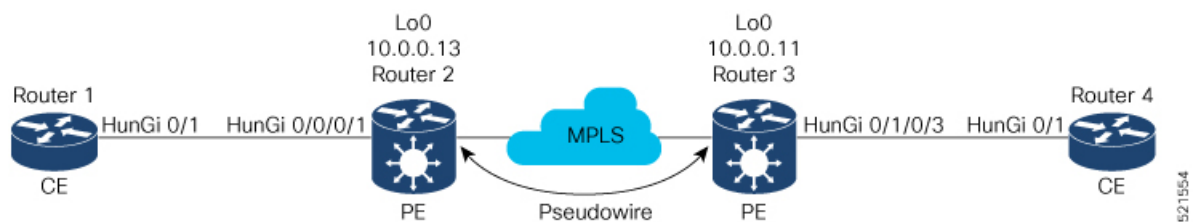
Pseudowire over MPLS or Ethernet-over-MPLS (EoMPLS) provides a tunneling mechanism for Ethernet traffic through an MPLS-enabled Layer 3 core network. PW over MPLS encapsulates Ethernet protocol data units (PDUs) using MPLS labels to forward them across the MPLS network.

### Limitations for Pseudowire over MPLS

- During Route Processor Failover (RPFO) events, if you have logging pseudowire (PW) configured under L2VPN:
  - The Cisco IOS XR software displays syslog messages indicating that LDP-signaled L2VPN VPWS pseudowire sessions temporarily go down and then come back up.
  - This session change does not impact traffic forwarding because the router maintains pseudowire forwarding programmed in hardware.
  - This ensures a continuous flow of traffic despite the syslog messages.

### Topology

Here is an example that showcases how the L2VPN traffic is transported using the PW over MPLS network.



- CEs are connected to PEs using the attachment circuit (AC).
- PW is configured on the PE devices to connect two PEs over an MPLS core network.

Consider a traffic flow from Router 1 to Router 4. Router 1 sends the traffic to Router 2 through the AC. Router 2 adds the MPLS PW label and sends it to Router 3 through the PW. Each PE needs to have an MPLS label in order to reach the loopback of the remote PE. This label, usually called the Interior Gateway Protocol (IGP) label, can be learned through the MPLS Label Distribution Protocol (LDP) or MPLS Traffic Engineering (TE).

One PE advertises the MPLS label to the other PE for PW identification. Router 3 identifies traffic with MPLS label and sends it to the AC connected to Router 4 after removing the MPLS label.

You can configure static or dynamic point-to-point connections.

# Configure Static Point-to-Point Connections Using Cross-Connect Circuits

This section describes how you can configure static point-to-point cross connects in a Layer 2 VPN.

## Requirements and Limitations

Before you can configure a cross-connect circuit in a Layer 2 VPN, ensure that the following requirements are met:

- The CE and PE routers are configured to operate in a network.
- The name of a cross-connect circuit is configured to identify a pair of PE routers and must be unique within the cross-connect group.
- A segment (an attachment circuit or pseudowire) is unique and can belong only to a single cross-connect circuit.
- A static virtual circuit local label is globally unique and can be used in only one pseudowire.

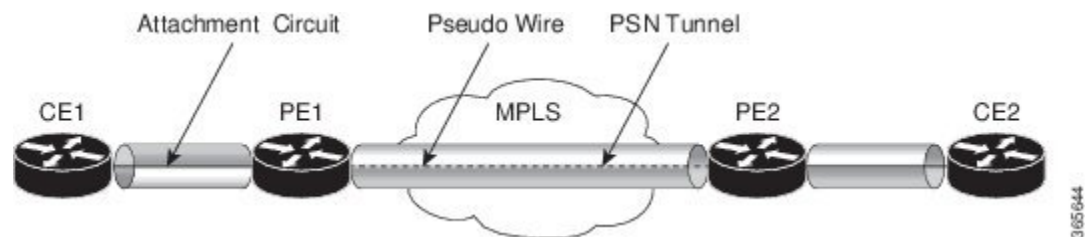


**Note** Static pseudowire connections do not use LDP for signaling.

## Topology

The following topology is used to configure static cross-connect circuits in a Layer 2 VPN.

**Figure 1: Static Cross-Connect Circuits in a Layer 2 VPN**



## Configuration

```
/* Configure PE1 */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group XCON1
Router(config-l2vpn-xc)# p2p xc1
Router(config-l2vpn-xc-p2p)# interface HundredGigEt0/1/0/0.1
Router(config-l2vpn-xc-p2p)# neighbor 10.0.0.3 pw-id 100
Router(config-l2vpn-xc-p2p-pw)# mpls static label local 50 remote 40
Router(config-l2vpn-xc-p2p-pw)# commit

/*Configure PE2 */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group XCON1
Router(config-l2vpn-xc)# p2p xc1
Router(config-l2vpn-xc-p2p)# interface HundredGigE0/2/0/0.4
Router(config-l2vpn-xc-p2p)# neighbor 10.0.0.4 pw-id 100
```

```
Router(config-l2vpn-xc-p2p-pw)# mpls static label local 40 remote 50
Router(config-l2vpn-xc-p2p-pw)# commit
```

### Running Configuration

```
/* On PE1 */
!
l2vpn
xconnect group XCON1
p2p xc1
    interface HundredGigE0/1/0/0.1
    neighbor ipv4 10.0.0.3 pw-id 100
    mpls static label local 50 remote 40
!

/* On PE2 */
!
l2vpn
xconnect group XCON2
p2p xc1
    interface HundredGigE0/2/0/0.4
    neighbor ipv4 10.0.0.4 pw-id 100
    mpls static label local 40 remote 50
!
```

### Verification

```
/* Verify the static cross connect on PE1 */
Router# show l2vpn xconnect
Tue Apr 12 20:18:02.971 IST
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
        SB = Standby, SR = Standby Ready, (PP) = Partially Programmed
```

XConnect			Segment 1		Segment 2	
Group	Name	ST	Description	ST	Description	ST
XCON1	xc1	UP	Hu0/1/0/0.1	UP	10.0.0.3 100	UP

```
/* Verify the static cross connect on PE2 */
```

```
Router# show l2vpn xconnect
Tue Apr 12 20:18:02.971 IST
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
        SB = Standby, SR = Standby Ready, (PP) = Partially Programmed
```

XConnect			Segment 1		Segment 2	
Group	Name	ST	Description	ST	Description	ST
XCON2	xc1	UP	Hu0/2/0/0.4	UP	10.0.0.4 100	UP

## Configure Dynamic Point-to-point Cross-Connects

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



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

## Configuration

```
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group vlan_grp_1
Router(config-l2vpn-xc)# p2p vlan1
Router(config-l2vpn-xc-p2p)# interface HunGigE 0/0/0/0.1
Router(config-l2vpn-xc-p2p)# neighbor 10.0.0.1 pw-id 1
Router(config-l2vpn-xc-p2p-pw)# commit
```

## Running Configuration

```
configure
l2vpn
  xconnect group vlan_grp_1
  p2p vlan1
  interface HunGigE 0/0/0/0.1
  neighbor 10.0.0.1 pw-id 1
!
```

# PW over MPLS Supported Modes

The PW over MPLS support these modes:

## Ethernet Port Mode

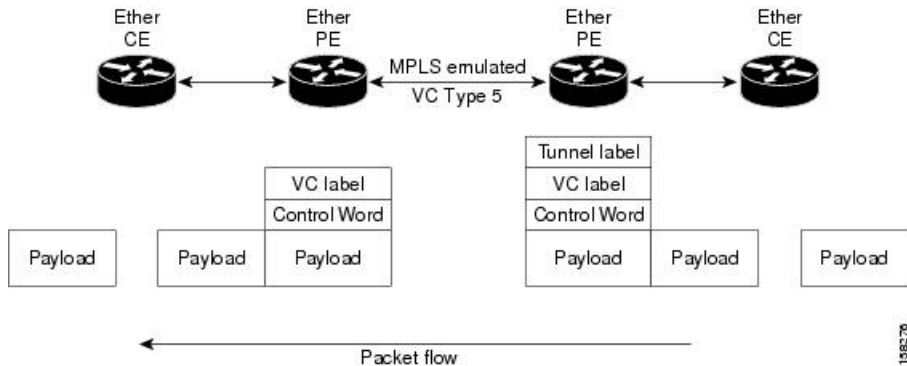
**Table 2: Feature History Table**

Feature Name	Release Information	Feature Description
Pseudowire VC Type 5	Release 7.3.15	With this feature, Ethernet port mode is supported for pseudowire over MPLS. The virtual connection (VC) type 5 is known as an Ethernet port-based PW. In this mode, both ends of a pseudowire are connected to Ethernet ports and allow a complete ethernet trunk to be transported. The ingress PE transports frames received on a main interface or subinterface. This feature nullifies the need for a dummy tag and reduces overhead. In addition, frame tagging is no longer necessary.

In Ethernet port mode, both ends of a pseudowire are connected to Ethernet ports. In this mode, the port is tunneled over the pseudowire. The ingress PE transports frames received on a main interface or after the subinterface tags are removed when the packet is received on a subinterface. The VLAN manipulation is transported over the type 5 PW, whether tagged or untagged.

This figure shows a sample ethernet port mode packet flow:

Figure 2: Ethernet Port Mode Packet Flow



### Configure Ethernet Port Mode

Perform this task to configure the Ethernet port mode.

```
/* PE1 configuration */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group grp1
Router(config-l2vpn-xc)# p2p xc1
Router(config-l2vpn-xc-p2p)# interface HundredGigE0/0/0/1.2
Router(config-l2vpn-xc-p2p)# neighbor 10.0.0.11 pw-id 222
Router(config-l2vpn-xc-p2p-pw)# commit

/* PE2 configuration */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group grp1
Router(config-l2vpn-xc)# p2p xc1
Router(config-l2vpn-xc-p2p)# interface HundredGigE0/1/0/3.2
Router(config-l2vpn-xc-p2p)# neighbor 10.0.0.13 pw-id 222
Router(config-l2vpn-xc-p2p-pw)# commit
```

### Running Configuration

This section shows the Ethernet port mode running configuration.

```
/* PE1 configuration */
l2vpn
xconnect group grp1
p2p xc1
  interface HundredGigE0/0/0/1.2
  neighbor 10.0.0.11 pw-id 222

/* PE2 configuration */
l2vpn
xconnect group grp1
p2p xc1
  interface HundredGigE0/1/0/3.2
  neighbor 10.0.0.13 pw-id 222
```

### Verification

Verify the Ethernet port mode configuration.

The PW type Ethernet indicates a VC type 5 PW.

```

Router# show l2vpn xconnect group grpl detail
Group grpl, XC xcl, state is up; Interworking none
AC: HundredGigE0/0/0/1.2, state is up
  Type VLAN; Num Ranges: 1
  VLAN ranges: [2, 2]
  MTU 1504; XC ID 0x840006; interworking none
  Statistics:
    packets: received 186, sent 38448
    bytes: received 12644, sent 2614356
    drops: illegal VLAN 0, illegal length 0
PW: neighbor 10.0.0.11, PW ID 222, state is up ( established )
  PW class not set, XC ID 0xc0000004
  Encapsulation MPLS, protocol LDP
  Source address 10.0.0.13
  PW type Ethernet, control word disabled, interworking none
  PW backup disable delay 0 sec
  Sequencing not set

PW Status TLV in use
  MPLS          Local          Remote
  -----
  Label          16026          16031
  Group ID       0x4000280       0x6000180
  Interface      HundredGigE0/0/0/1.2  HundredGigE0/1/0/3.2
  MTU            1504          1504
  Control word   disabled       disabled
  PW type        Ethernet       Ethernet
  VCCV CV type   0x2          0x2
                  (LSP ping verification)  (LSP ping verification)
  VCCV CC type   0x6          0x6
                  (router alert label)  (router alert label)
                  (TTL expiry)        (TTL expiry)
  -----
Incoming Status (PW Status TLV):
  Status code: 0x0 (Up) in Notification message
Outgoing Status (PW Status TLV):
  Status code: 0x0 (Up) in Notification message
MIB cpwVcIndex: 3221225476
Create time: 30/03/2021 16:30:58 (21:31:00 ago)
Last time status changed: 30/03/2021 16:36:42 (21:25:16 ago)
Statistics:
  packets: received 38448, sent 186
  bytes: received 2614356, sent 12644

```

## VLAN Mode

**Table 3: Feature History Table**

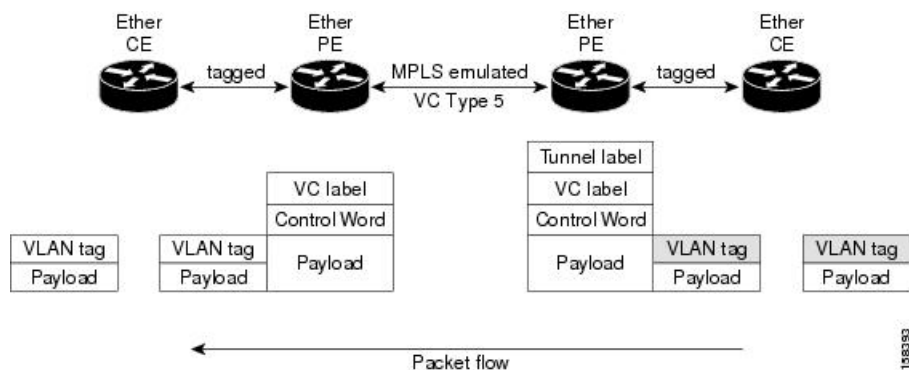
Feature Name	Release Information	Feature Description

Pseudowire VC Type 4	Release 7.3.15	With this feature, VLAN mode is supported for pseudowire over MPLS. A virtual connection (VC) type 4 is the VLAN-based PW. The ingress PE does not remove the incoming VLAN tags that are to be transported over the PW. VC type 4 inserts an extra dummy tag with VLAN 0 onto the frame which is removed on the other side. This mode helps the service provider to segregate traffic for each customer based on the VLAN.
----------------------	----------------	---

In VLAN mode, each VLAN on a customer-end to provider-end link can be configured as a separate L2VPN connection using virtual connection (VC) type 4. In VLAN mode, each VLAN on a customer-end to provider-end link can be configured as a separate L2VPN connection using virtual connection (VC) type 4. VLAN-based (VC Type 4) pseudowires ensure a VLAN tag is transported over the pseudowire by pushing a dummy tag at the attachment circuit ingress. If the rewrite rule pushes two or more tags, a dummy tag is not needed because these VLAN tags are transported over the pseudowire. On the remote router, the dummy tag, if added, is removed before egress.

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

**Figure 3: VLAN Mode Packet Flow**



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



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

### Limitation

On PW imposition PE, the pushed dummy VLAN Tag Tag Protocol Identifier (TPID) is copied from the TPID of the innermost VLAN tag popped on the ingress L2 interface where traffic is received from. If there is no VLAN tag popped on the L2 interface, the TPID on the dummy VLAN is 0x8100.

On the disposition PE, if the egress VLAN tag push is configured on the egress L2 interface, the innermost pushed VLAN tag TPID is copied from the TPID of the dummy VLAN tag. If there is no egress VLAN push configured on the egress L2 interface, the dummy VLAN tag is discarded.



## Configure VLAN Mode

Perform this task to configure VLAN mode.

```
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# pw-class VLAN
Router(config-l2vpn-pwc)# encapsulation mpls
Router(config-l2vpn-pwc-mpls)# transport-mode vlan
Router(config-l2vpn-pwc-mpls)# exit
Router(config-l2vpn-pwc)# exit
Router(config-l2vpn)# xconnect group grp1
Router(config-l2vpn-xc)# p2p xc1
Router(config-l2vpn-xc-p2p)# neighbor 10.0.0.11 pw-id 222
Router(config-l2vpn-xc-p2p-pw)# pw-class VLAN
Router(config-l2vpn-xc-p2p-pw)# commit
```

## Running Configuration

This section shows the VLAN mode running configuration.

```
l2vpn
pw-class VLAN
  encapsulation mpls
  transport-mode vlan
  !
  !
xconnect group grp1
p2p xc1
  neighbor 10.0.0.11 pw-id 222
  pw-class VLAN
  !
  !
  !
  !
```

## Verification

Verify the VLAN mode configuration.

The PW type Ethernet VLAN indicates a type 4 PW.

```
Router# show l2vpn xconnect group grp1 detail | i " PW type"
PW type Ethernet VLAN, control word disabled, interworking none
      PW type      Ethernet VLAN      Ethernet VLAN
```

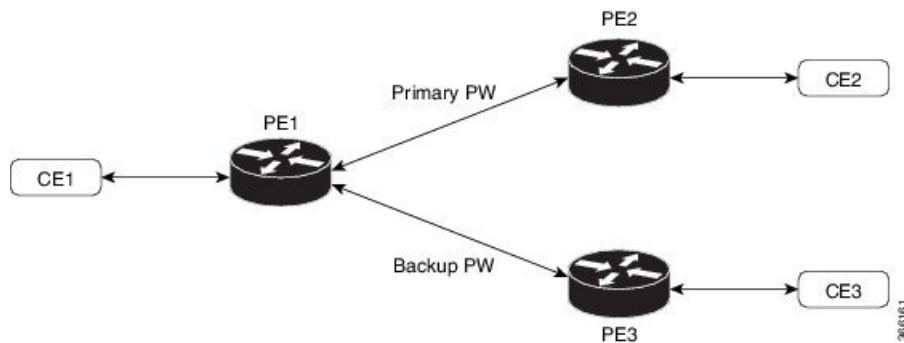
# VLAN Passthrough Mode

Configure the **transport mode vlan passthrough** command under the pw-class to negotiate a virtual connection (VC)-type 4 (Ethernet VLAN) PW, which transports whatever comes out of the AC after the VLAN tag manipulation specified by the **rewrite** command. The VLAN tag manipulation on the EFP ensures that there is at least one VLAN tag left on the frame because you need a VLAN tag on the frame if there are VC-type 4 PWs. No dummy tag 0 is added to the frame when you use the **transport mode vlan passthrough** command.

## Pseudowire Redundancy

The Pseudowire Redundancy feature allows you to configure a redundant pseudowire that backs up the primary pseudowire. When the primary pseudowire fails, the PE router switches to the redundant pseudowire. You can elect to have the primary pseudowire resume operation after it becomes functional. The primary pseudowire fails when the PE router fails or when there is a network outage.

**Figure 4: Pseudowire Redundancy**



### Forcing a Manual Switchover to the Backup Pseudowire

To force the router to switch over to the backup or switch back to the primary pseudowire, use the **l2vpn switchover** command in EXEC mode.

A manual switchover is made only if the peer specified in the command is actually available and the cross-connect moves to the fully active state when the command is entered.

## Configure Pseudowire Redundancy

This section describes how you can configure pseudowire redundancy.

You must consider the following restrictions while configuring the Pseudowire Redundancy feature:

- 2000 active and 2000 backup PWs are supported.
- Only MPLS LDP is supported.

```

/* Configure PW on PE1 */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group XCON1
Router(config-l2vpn-xc)# p2p xc1
Router(config-l2vpn-xc-p2p)# interface HundredGigE0/1/0/0.1
Router(config-l2vpn-xc-p2p)# neighbor ipv4 172.16.0.1 pw-id 1
Router(config-l2vpn-xc-p2p-pw)# backup neighbor 192.168.0.1 pw-id 1
Router(config-l2vpn-xc-p2p-pw-backup)# commit

/* Configure PW on PE2 */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group XCON1
Router(config-l2vpn-xc)# p2p xc1
Router(config-l2vpn-xc-p2p)# interface HundredGigE0/1/0/0.1
Router(config-l2vpn-xc-p2p)# neighbor ipv4 10.0.0.1 pw-id 1
Router(config-l2vpn-xc-p2p-pw)# commit

```

```

/* Configure PW on PE3 */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group XCON1
Router(config-l2vpn-xc)# p2p xc1
Router(config-l2vpn-xc-p2p)# interface HundredGigE0/1/0/0.1
Router(config-l2vpn-xc-p2p)# neighbor ipv4 10.0.0.1 pw-id 1
Router(config-l2vpn-xc-p2p-pw)# commit

```

## Running Configuration

```

/* On PE1 */
!
l2vpn
xconnect group XCON1
p2p XCON1_P2P2
interface HundredGigE 0/1/0/0.1
neighbor ipv4 172.16.0.1 pw-id 1
backup neighbor 192.168.0.1 pw-id 1
!

/* On PE2 */
!
l2vpn
xconnect group XCON1
p2p XCON1_P2P2
interface HundredGigE 0/1/0/0.1
neighbor ipv4 10.0.0.1 pw-id 1
!

/* On PE3 */
!
l2vpn
xconnect group XCON1
p2p XCON1_P2P2
interface HundredGigE 0/1/0/0.1
neighbor ipv4 10.0.0.1 pw-id 1
!

```

## Verification

Verify that the configured pseudowire redundancy is up.

```

/* On PE1 */

Router#show l2vpn xconnect group XCON_1
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
        SB = Standby, SR = Standby Ready, (PP) = Partially Programmed

```

XConnect			Segment 1		Segment 2		
Group	Name	ST	Description	ST	Description	ST	
XCON_1	XCON1_P2P2	UP	Hu0/1/0/0.1	UP	172.16.0.1	UP	
					Backup		
					192.168.0.1	SB	

```

/* On PE2 */

Router#show l2vpn xconnect group XCON_1

```

Tue Jan 17 15:36:12.327 UTC

Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,  
SB = Standby, SR = Standby Ready, (PP) = Partially Programmed

XConnect			Segment 1		Segment 2		
Group	Name	ST	Description	ST	Description		ST
XCON_1	XCON1_P2P2	UP	BE100.1	UP	10.0.0.1	1000	UP

/\* On PE3 \*/

Router#show l2vpn xconnect group XCON\_1

Tue Jan 17 15:38:04.785 UTC

Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,  
SB = Standby, SR = Standby Ready, (PP) = Partially Programmed

XConnect			Segment 1		Segment 2		
Group	Name	ST	Description	ST	Description		ST
XCON_1	XCON1_P2P2	DN	BE100.1	UP	10.0.0.1	1000	SB

Router#show l2vpn xconnect summary

Number of groups: 3950

Number of xconnects: 3950

Up: 3950 Down: 0 Unresolved: 0 Partially-programmed: 0

AC-PW: 3950 AC-AC: 0 PW-PW: 0 Monitor-Session-PW: 0

Number of Admin Down segments: 0

Number of MP2MP xconnects: 0

Up 0 Down 0

Advertised: 0 Non-Advertised: 0

Number of CE Connections: 0

Advertised: 0 Non-Advertised: 0

Backup PW:

Configured : 3950

UP : 0

Down : 0

Admin Down : 0

Unresolved : 0

Standby : 3950

Standby Ready: 0

Backup Interface:

Configured : 0

UP : 0

Down : 0

Admin Down : 0

Unresolved : 0

Standby : 0

## Inter-AS Mode

**Table 4: Feature History Table**

Feature Name	Release Information	Feature Description
--------------	---------------------	---------------------

Inter-AS Mode for L2VPN Pseudowire	Release 7.3.15	Inter-AS is a peer-to-peer type that allows VPNs to operate through multiple providers or multi-domain networks using L2VPN cross-connect. This mode allows VPLS autodiscovery to operate across multiple BGP autonomous systems and enables service providers to offer end-to-end VPN connectivity over different geographical locations.
------------------------------------	----------------	--

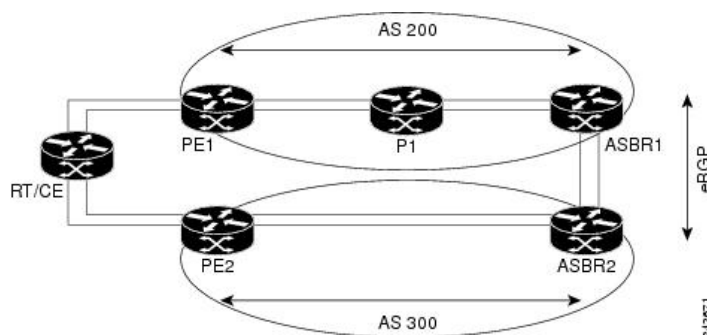
An autonomous system (AS) is a single network or group of networks that is controlled by a common system administration group and uses a single, clearly defined routing protocol.

As VPNs grow, their requirements expand. In some cases, VPNs need to reside on different autonomous systems in different geographic areas. In addition, some VPNs need to extend across multiple service providers (overlapping VPNs). Regardless of the complexity and location of the VPNs, the connection between autonomous systems must be seamless.

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

The following figure illustrates MPLS over Inter-AS with a basic double AS topology with iBGP/LDP in each AS.

**Figure 5: EoMPLS over Inter-AS: Basic Double AS Topology**



## Configure Inter-AS Mode

Perform this task to configure Inter-AS mode:

```
/* PE1 Configuration */
Router# configure
Router(config)# mpls ldp
Router(config-ldp)# router-id 10.0.0.1
Router(config-ldp)# interface HundredGigE0/2/0/3
Router(config-ldp-if)# exit
Router(config-ldp)# router bgp 100
Router(config-bgp)# bgp router-id 10.0.0.1
Router(config-bgp)# address-family l2vpn vpls-vpws
Router(config-bgp-af)# neighbor 172.16.0.1
Router(config-bgp-nbr)# remote-as 200
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family l2vpn vpls-vpws
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# exit
Router(config-bgp)# exit
Router(config)# l2vpn
```

```

Router(config-l2vpn)# xconnect group gr1
Router(config-l2vpn-xc)# mp2mp mp1
Router(config-l2vpn-xc-mp2mp)# vpn-id 100
Router(config-l2vpn-xc-mp2mp)# l2-encapsulation vlan
Router(config-l2vpn-xc-mp2mp)# autodiscovery bgp
Router(config-l2vpn-xc-mp2mp-ad)# rd auto
Router(config-l2vpn-xc-mp2mp-ad)# route-target 2.2.2.2:100
Router(config-l2vpn-xc-mp2mp-ad)# signaling-protocol bgp
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 1
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface HunGigE0/1/0/1.1 remote-ce-id 2
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface HunGigE0/1/0/1.1 remote-ce-id 3
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit

```

```

/* PE2 Configuration */
Router# configure
Router(config)# mpls ldp
Router(config-ldp)# router-id 172.16.0.1
Router(config-ldp)# interface HundredGigE0/3/0/0
Router(config-ldp-if)# exit
Router(config-ldp)# router bgp 100
Router(config-bgp)# bgp router-id 172.16.0.1
Router(config-bgp)# address-family l2vpn vpls-vpws
Router(config-bgp-af)# neighbor 10.0.0.1
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family l2vpn vpls-vpws
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# exit
Router(config-bgp)# exit
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group gr1
Router(config-l2vpn-xc)# mp2mp mp1
Router(config-l2vpn-xc-mp2mp)# vpn-id 100
Router(config-l2vpn-xc-mp2mp)# l2-encapsulation vlan
Router(config-l2vpn-xc-mp2mp)# autodiscovery bgp
Router(config-l2vpn-xc-mp2mp-ad)# rd auto
Router(config-l2vpn-xc-mp2mp-ad)# route-target 2.2.2.2:100
Router(config-l2vpn-xc-mp2mp-ad)# signaling-protocol bgp
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 2
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface HunGigE0/1/0/2.1 remote-ce-id 3
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface HunGigE0/1/0/2.2 remote-ce-id 1
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit

```

## Running Configuration

This section shows the Inter-AS running configuration.

```

/* PE1 Configuration */
mpls ldp
router-id 10.0.0.1
interface HundredGigE0/2/0/3
!
router bgp 100
bgp router-id 10.0.0.1
address-family l2vpn vpls-vpws
neighbor 172.16.0.1
remote-as 200
update-source Loopback0
address-family l2vpn vpls-vpws
!
l2vpn

```

```

xconnect group gr1
mp2mp mp1
  vpn-id 100
  l2-encapsulation vlan
  autodiscovery bgp
  rd auto
  route-target 2.2.2.2:100
  signaling-protocol bgp
  ce-id 1
    interface HunGigE0/1/0/1.1 remote-ce-id 2
    interface HunGigE0/1/0/1.2 remote-ce-id 3

/* PE2 Configuration */
mpls ldp
  router-id 172.16.0.1
  interface HundredGigE0/3/0/0
  !
router bgp 100
  bgp router-id 172.16.0.1
  address-family l2vpn vpls-vpws
  neighbor 10.0.0.1
  remote-as 100
  update-source Loopback0
  address-family l2vpn vpls-vpws
!
l2vpn
xconnect group gr1
mp2mp mp1
  vpn-id 100
  l2-encapsulation vlan
  autodiscovery bgp
  rd auto
  route-target 2.2.2.2:100
  signaling-protocol bgp
  ce-id 2
    interface HunGigE0/1/0/2.1 remote-ce-id 3
    interface HunGigE0/1/0/2.2 remote-ce-id 1

```

## Virtual Circuit Connection Verification on L2VPN

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

Two types of packets can arrive at the disposition egress:

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

Cisco 8000 series router supports Label Switched Path (LSP) VCCV Type 1, which uses an inband control word if enabled during signaling. The VCCV echo reply is sent as IPv4 that is the reply mode in IPv4. The reply is forwarded as IP, MPLS, or a combination of both.

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

# Preferred tunnel path

Preferred tunnel path is a configuration mechanism that:

- influences the selection of the transport path for Layer 2 traffic within service provider networks,
- maps pseudowires to specific traffic engineering tunnels, and
- facilitates targeted configuration of network traffic paths through specific tunnels.

With the preferred tunnel path feature:

- You can map pseudowires to specific traffic engineering tunnels.
- Attachment circuits are cross-connected to specific MPLS traffic engineering tunnel interfaces instead of remote PE router IP addresses (reachable using IGP or LDP).
- The traffic engineering tunnel transports the L2 traffic between two PE routers, with the headend starting at the imposition PE router and the tailend terminating on the disposition PE router.

## Benefits

- Optimizes network resource utilization and performance when you set an explicit path on the PE router to direct traffic flow to a specific destination PE router.
- Supports fallback enable option.
- Enables you to troubleshoot MPLS networks with its MPLS OAM capabilities.

## Restrictions

The preferred tunnel path configuration applies only to MPLS encapsulation.

## Configure preferred tunnel path

This procedure provides the configuration for preferred tunnel path.

### Procedure

---

#### Step 1 **configure**

##### **Example:**

```
Router# configure
```

Enters the global configuration mode.

#### Step 2 **l2vpn**

##### **Example:**

```
Router(config)# l2vpn
```



Enters the L2VPN configuration mode.

**Step 3**     **pw-classpath**

**Example:**

```
Router(config-l2vpn)# pw-class PATH1
```

Defines the pseudowire class, `PATH1`, to configure tunnel preferences.

**Step 4**     **encapsulation mpls**

**Example:**

```
Router(config-l2vpn-pwc)# encapsulation mpls
```

Specifies MPLS as the encapsulation type for the pseudowire.

**Step 5**     **preferred-path interface tunnel-te valuefallback disable**

**Example:**

```
Router(config-l2vpn-pwc-mpls)# preferred-path interface tunnel-te1 fallback disable
```

Here, **interface** *tunnel-te1* specifies the preferred MPLS TE tunnel and **fallback disable** ensures that no alternative path is used if the preferred tunnel is unavailable.

**Step 6**     **commit**

**Example:**

```
Router(config-l2vpn-pwc-mpls)# commit
```

Saves and applies the configuration changes.

**Step 7**     **exit**

**Example:**

```
Router(config-l2vpn-pwc-mpls)# exit
Router(config-l2vpn-pwc)# exit
Router(config-l2vpn)# exit
Router(config)# exit
```

Exits the global configuration mode.

---

## Configure MPLS-TE tunnel for VPLS

This procedure provides the MPLS-TE tunnel configuration for VPLS.

### Procedure

---

**Step 1**     **configure**

**Example:**

```
Router# configure
```

Enters the global configuration mode.

**Step 2**     **interface** *interface-name***Example:**

```
Router(config)# interface tunnel-te1
```

Enters the configuration mode for the MPLS-TE tunnel interface.

**Step 3**     **ipv4 unnumbered Loopback0****Example:**

```
Router(config-if)# ipv4 unnumbered Loopback0
```

Sets the tunnel to use an IPv4 unnumbered loopback interface.

**Step 4**     **signalled-bandwidth** *value***Example:**

```
Router(config-if)# signalled-bandwidth 50
```

Specifies the signaled bandwidth for the tunnel.

**Step 5**     **destination** *ip-address***Example:**

```
Router(config-if)# destination 10.12.12.12
```

Defines the destination IP address for the tunnel.

**Step 6**     **path-option** *value* **explicit name** *name***Example:**

```
Router(config-if)# path-option 1 explicit name FC1
```

Specifies the explicit path option.

**Step 7**     **exit****Example:**

```
Router(config-if)# exit
```

Exits the tunnel interface configuration mode.

**Step 8**     **commit****Example:**

```
Router(config)# commit
```

Saves and applies the configuration changes.

**Step 9**     **exit****Example:**

```
Router(config)# exit
```

Exits the global configuration mode and returns to the EXEC mode.

---

## Configure VPLS over preferred TE tunnel

This procedure provides the VPLS configuration over a preferred TE tunnel.

### Procedure

---

- Step 1**      **configure**  
**Example:**  
`Router# configure`  
Enters the global configuration mode.
- Step 2**      **interface** *interface-name* **l2transport**  
**Example:**  
`Router(config)# interface FourHundredGigE0/0/0/0.1 l2transport`  
Defines the Layer 2 transport on the physical or subinterface that will carry VPLS traffic.
- Step 3**      **encapsulation dot1q** *value*  
**Example:**  
`Router(config-subif)# encapsulation dot1q 100`  
Specifies the VLAN encapsulation for the interface.
- Step 4**      **rewrite ingress tag pop** *value* **symmetric**  
**Example:**  
`Router(config-subif)# rewrite ingress tag pop 1 symmetric`  
Specifies the rewrite rule to pop one VLAN tag symmetrically for ingress traffic.
- Step 5**      **exit**  
**Example:**  
`Router(config-subif)# exit`  
Exits the subinterface configuration mode.
- Step 6**      **l2vpn**  
**Example:**  
`Router(config)# l2vpn`  
Enters the L2VPN configuration mode.
- Step 7**      **pw-class** *name*  
**Example:**  
`Router(config-l2vpn)# pw-class c`  
Creates the pseudowire class for the VPLS pseudowire.
- Step 8**      **encapsulation mpls**

**Example:**

```
Router(config-l2vpn-pwc)# encapsulation mpls
```

Specifies MPLS encapsulation for the pseudowire.

**Step 9 control-word****Example:**

```
Router(config-l2vpn-pwc-mpls)# control-word
```

Enables control word for the pseudowire.

**Step 10 load-balancing****Example:**

```
Router(config-l2vpn-pwc-mpls)# load-balancing
```

Enables the load balancing configuration mode.

**Step 11 flow-label both****Example:**

```
Router(config-l2vpn-pwc-mpls-load-bal)# flow-label both
```

Enables flow-label based load balancing for both ingress and egress traffic.

Exits the load balancing configuration mode.

**Step 12 preferred path interface *interface-name*****Example:**

```
Router(config-l2vpn-pwc-mpls)# preferred-path interface tunnel-te1
```

Configures the preferred MPLS-TE tunnel path for the pseudowire.

**Step 13 exit****Example:**

```
Router(config-l2vpn-pwc-mpls)# exit
```

```
Router(config-l2vpn-pwc)# exit
```

Exits the pseudowire class configuration mode.

**Step 14 brige group *group-name* bridge-domain *domain-name*****Example:**

```
Router(config-l2vpn)# bridge group bg bridge-domain bd
```

Defines the VPLS bridge group and bridge domain.

**Step 15 interface *interface-name*****Example:**

```
Router(config-l2vpn-bg-bd)# interface FourHundredGigE0/0/0/0.1
```

```
Router(config-l2vpn-bg-bd-ac)# exit
```

Associates the access interface with the bridge domain.

**Step 16 neighbor *ip-address* pw-id *value*****Example:**

```
Router(config-l2vpn-bg-bd) # neighbor 10.12.12.12 pw-id 100
```

Defines a pseudowire neighbor for the bridge domain.

### Step 17 **pw-class class-name**

#### Example:

```
Router(config-l2vpn-bg-bd-pw) # pw-class c
Router(config-l2vpn-bg-bd-pw) # exit
Router(config-l2vpn-bg-bd) # exit
Router(config-l2vpn) # exit
```

Associates the pseudowire class with the neighbor pseudowire.

### Step 18 **commit**

#### Example:

```
Router(config) # commit
```

Saves and applies the configuration changes.

## Verification

```
RP/0/RP0/CPU0:r1#show l2vpn bridge-domain detail
Wed Apr 20 17:53:26.232 UTC
Legend: pp = Partially Programmed.
Bridge group: bg, bridge-domain: bd, id: 0, state: up, ShgId: 0, MSTi: 0
  Coupled state: disabled
  VINE state: Default
  MAC learning: enabled
  MAC withdraw: enabled
    MAC withdraw for Access PW: enabled
    MAC withdraw sent on: bridge port up
    MAC withdraw relaying (access to access): disabled
  Flooding:
    Broadcast & Multicast: enabled
    Unknown unicast: enabled
  MAC aging time: 300 s, Type: inactivity
  MAC limit: 131072, Action: none, Notification: syslog
  MAC limit reached: no, threshold: 75%
  MAC port down flush: enabled
  MAC Secure: disabled, Logging: disabled
  Split Horizon Group: none
  E-Tree: Root
  Dynamic ARP Inspection: disabled, Logging: disabled
  IP Source Guard: disabled, Logging: disabled
  DHCPv4 Snooping: disabled
  DHCPv4 Snooping profile: none
  IGMP Snooping: disabled
  IGMP Snooping profile: none
  MLD Snooping profile: none
  Storm Control: disabled
  Bridge MTU: 1500
  MIB cvplsConfigIndex: 1
  Filter MAC addresses:
  P2MP PW: disabled
  Multicast Source: Not Set
  Create time: 20/04/2022 17:37:30 (00:15:55 ago)
  No status change since creation
  ACs: 1 (1 up), VFIs: 0, PWs: 1 (1 up), PBBs: 0 (0 up), VNIs: 0 (0 up)
```

## List of ACs:

```

AC: FourHundredGigE0/0/0/0, state is up
Type Ethernet
MTU 1500; XC ID 0x1; interworking none
MAC learning: enabled
Flooding:
  Broadcast & Multicast: enabled
  Unknown unicast: enabled
MAC aging time: 300 s, Type: inactivity
MAC limit: 131072, Action: none, Notification: syslog
MAC limit reached: no, threshold: 75%
MAC port down flush: enabled
MAC Secure: disabled, Logging: disabled
Split Horizon Group: none
E-Tree: Root
Dynamic ARP Inspection: disabled, Logging: disabled
IP Source Guard: disabled, Logging: disabled
DHCPv4 Snooping: disabled
DHCPv4 Snooping profile: none
IGMP Snooping: disabled
IGMP Snooping profile: none
MLD Snooping profile: none
Storm Control: bridge-domain policer
Static MAC addresses:
Statistics:
  packets: received 0 (multicast 0, broadcast 0, unknown unicast 0, unicast 0), sent
0
  bytes: received 0 (multicast 0, broadcast 0, unknown unicast 0, unicast 0), sent 0
  MAC move: 0
Storm control drop counters:
  packets: broadcast 0, multicast 0, unknown unicast 0
  bytes: broadcast 0, multicast 0, unknown unicast 0
Dynamic ARP inspection drop counters:
  packets: 0, bytes: 0
IP source guard drop counters:
  packets: 0, bytes: 0
PD System Data: Learn key: 0

```

## List of Access PWs:

```

PW: neighbor 10.12.12.12, PW ID 100, state is up ( established )
PW class c, XC ID 0xa0000001
Encapsulation MPLS, protocol LDP
Source address 10.10.10.10
PW type Ethernet, control word enabled, interworking none
PW backup disable delay 0 sec
Sequencing not set
Preferred path Active : tunnel-te1, Statically configured, fallback enabled
Ignore MTU mismatch: Disabled
Transmit MTU zero: Disabled
Tunnel : Up

```

## PW Status TLV in use

MPLS	Local	Remote
Label	24000	24000
Group ID	0x0	0x0
Interface	Access PW	Access PW
MTU	1500	1500
Control word	enabled	enabled
PW type	Ethernet	Ethernet
VCCV CV type	0x2	0x2
	(LSP ping verification)	(LSP ping verification)
VCCV CC type	0x7	0x7
	(control word)	(control word)
	(router alert label)	(router alert label)

```

(TTL expiry)
-----
Incoming Status (PW Status TLV):
  Status code: 0x0 (Up) in Notification message
  MIB cpwVcIndex: 2684354561
  Create time: 20/04/2022 17:37:30 (00:15:55 ago)
  Last time status changed: 20/04/2022 17:53:22 (00:00:04 ago)
  MAC withdraw messages: sent 0, received 0
  Forward-class: 0
  Static MAC addresses:
  Statistics:
    packets: received 0 (unicast 0), sent 0
    bytes: received 0 (unicast 0), sent 0
    MAC move: 0
  Storm control drop counters:
    packets: broadcast 0, multicast 0, unknown unicast 0
    bytes: broadcast 0, multicast 0, unknown unicast 0
  MAC learning: enabled
  Flooding:
    Broadcast & Multicast: enabled
    Unknown unicast: enabled
  MAC aging time: 300 s, Type: inactivity
  MAC limit: 131072, Action: none, Notification: syslog
  MAC limit reached: no, threshold: 75%
  MAC port down flush: enabled
  MAC Secure: disabled, Logging: disabled
  Split Horizon Group: none
  E-Tree: Root
  DHCPv4 Snooping: disabled
  DHCPv4 Snooping profile: none
  IGMP Snooping: disabled
  IGMP Snooping profile: none
  MLD Snooping profile: none
  Storm Control: bridge-domain policer
(TTL expiry)
-----

```

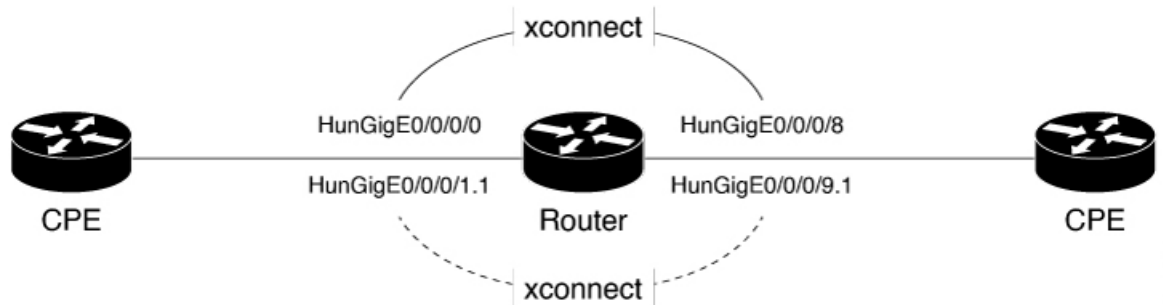
## Configure Local Switching Between Attachment Circuits

**Table 5: Feature History Table**

Feature Name	Release Information	Feature Description
Support of Tagged or Untagged VLAN on Physical and Bundle AC with VLAN Rewrite	Release 7.3.15	This feature supports tagged or untagged VLAN on physical and bundle interfaces. The tagged VLAN allows you to send and receive the traffic for multiple VLANs whereas, the untagged VLAN allows you to send and receive the traffic for a single VLAN. The multiple VLANs are used to differentiate traffic streams so that the traffic can be split across different services.

Local switching involves the exchange of L2 data from one attachment circuit (AC) to the other. The two ports configured in a local switching connection form an attachment circuit (AC). A local switching connection works like a bridge domain that has only two bridge ports, where traffic enters from one port of the local connection and leaves through the other.

Figure 6: Local Switching Between Attachment Circuits



521698

These are some of the characteristics of Layer 2 local switching:

- Because there is no bridging involved in a local connection, there is neither MAC learning nor flooding.
- ACs in a local connection are not in the UP state if the interface state is DOWN.
- Local switching ACs utilize a full variety of Layer 2 interfaces, including Layer 2 trunk (main) interfaces, bundle interfaces, and Ethernet Flow Points (EFPs).
- Same-port local switching allows you to switch Layer 2 data between two circuits on the same interface.

### Configuration

To configure an AC-AC point-to-point cross connect, complete the following configuration:

- Create Layer 2 interfaces.
- Create a cross-connect group and point-to-point connection.
- Attach the Layer 2 interfaces to point-to-point connection.

```
/* Configure L2 transport and encapsulation on the VLAN sub-interfaces */
Router# configure
Router(config)# interface HunGigE 0/0/0/1.1 l2transport
Router(config-subif)# encapsulation dot1q 5
Router(config-subif)# exit
Router(config)# interface HunGigE 0/0/0/9.1 l2transport
Router(config-subif)# encapsulation dot1q 5
Router(config-subif)# commit
```

```
/* Configure local switching on the VLAN sub-interfaces */
Router(config)# l2vpn
Router(config-l2vpn-xc)# p2p XCON1_P2P1
Router(config-l2vpn-xc-p2p)# interface HunGigE0/0/0/1.1
Router(config-l2vpn-xc-p2p)# interface HunGigE0/0/0/9.1
Router(config-l2vpn-xc-p2p)# commit
Router(config-l2vpn-xc-p2p)# exit
```



## Running Configuration

```
configure

interface HunGigE 0/0/0/1.1 l2transport
 encapsulation dot1q 5
!
interface HunGigE 0/0/0/9.1 l2transport
 encapsulation dot1q 5
!

l2vpn
 p2p XCON1_P2P1
  interface HunGigE0/0/0/1.1
  interface HunGigE0/0/0/9.1
  !
!
!
```

## Verification

- Verify if the configured cross-connect is UP

```
router# show l2vpn xconnect brief
```

```
Locally Switching
```

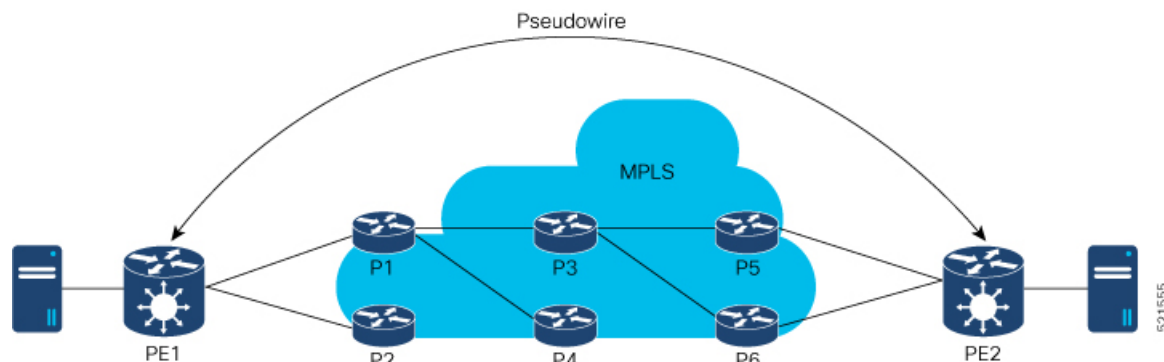
Like-to-Like	UP	DOWN	UNR
EFP	1	0	0
Total	1	0	0
Total	1	0	0

```
Total: 1 UP, 0 DOWN, 0 UNRESOLVED
```

# MPLS PW Traffic Load Balancing on P Router

When an L2VPN PE needs to send a frame over an MPLS PW, the Ethernet frame is encapsulated into an MPLS frame with one or more MPLS labels; there is at least one PW label and perhaps an IGP label to reach the remote PE.

The MPLS frame is transported by the MPLS network to the remote L2VPN PE. There are typically multiple paths to reach the destination PE:



PE1 can choose between P1 and P2 as the first MPLS P router towards PE2. If P1 is selected, P1 then chooses between P3 and P4, and so on. The available paths are based on the IGP topology and the MPLS TE tunnel path.

MPLS service providers prefer to have all links equally utilized rather than one congested link with other underutilized links. This goal is not always easy to achieve because some PWs carry much more traffic than others and because the path taken by a PW traffic depends upon the hashing algorithm used in the core. Multiple high bandwidth PWs might be hashed to the same links, which creates congestion.

A very important requirement is that all packets from one flow must follow the same path. Otherwise, this leads to out-of-order frames, which might impact the quality or the performance of the applications.

Use the following methods to load balance the MPLS PW traffic:

## Load Balance MPLS PW Traffic using Control-Word and Flow-Label

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
Load Balance MPLS PW Traffic using Flow-Label	Release 7.3.15	<p>The flow-label provides the capability to identify individual flows within a pseudowire and provides routers the ability to use these flows to load balance traffic. Individual flows are determined by the hashing algorithm configured under L2VPN. Similar packets with the same source and destination addresses are all said to be in the same flow. A flow-label is created based on indivisible packet flows entering a pseudowire and is inserted as the lowermost label in the packet. Routers can use the flow-label for load balancing which provides a better traffic distribution across ECMP paths or link-bundled paths in the core.</p> <p>The <b>flow-label</b> keyword is added.</p>

Load Balance MPLS PW Traffic using Control-Word	Release 7.3.15	<p>This feature allows the router to correctly identify the Ethernet PW packet over an IP packet, thus preventing the selection of wrong equal-cost multipath (ECMP) path for the packet that leads to the misordering of packets. This feature inserts the control word keyword immediately after the MPLS label to separate the payload from the MPLS label over a PW. The control word carries layer 2 control bits and enables sequencing.</p> <p>The <b>control-word</b> keyword is added.</p>
---	----------------	---

### Load balancing using Control-Word

If the MPLS packet contains the MAC address that starts with 0x4 or 0x6, a label switching router (LSR) misidentifies the Ethernet PW packet as an IP packet. The router considers that there is an IPv4 or IPv6 packet inside the MPLS packet and tries to load balance based on a hash of the source and destination IPv4 or IPv6 addresses extracted from the frame. This leads to the selection of the wrong equal-cost multipath (ECMP) path for the packet, leading to the misordering of packets.

This must not apply to an Ethernet frame that is encapsulated and transported over a PW because the destination MAC address considers the bottom label.

To overcome this issue, use the **control-word** keyword under a pw-class that is attached to a point-to-point PW. The control word is inserted immediately after the MPLS labels. Pseudowire over MPLS also, known as Ethernet over MPLS (EoMPLS), allows you to tunnel two L2VPN Provider Edge (PE) devices to transport L2VPN traffic over an MPLS cloud. This feature uses MPLS labels to transport data over the PW. The two L2VPN PEs are typically connected at two different sites with an MPLS core between them. This feature allows you to migrate legacy ATM and Frame Relay services to MPLS or IP core without interrupting the existing services.

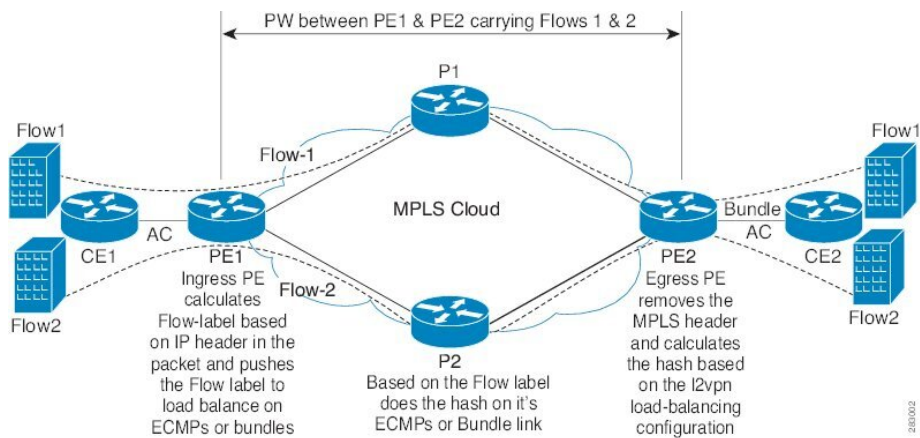
### Load balancing using Flow-Label

Routers typically load balance traffic based on the lowermost label in the label stack which is the same label for all flows on a given pseudowire. This can lead to asymmetric load balancing. The flow, in this context, refers to a sequence of packets that have the same source and destination pair. The packets are transported from a source provider edge (PE) to a destination PE.

The flow-label provides the capability to identify individual flows within a pseudowire and provides routers the ability to use these flows to load balance traffic. A flow-label is created based on individual packet flows entering a pseudowire and is inserted as the lowermost label in the packet. Routers can use the flow-label for load balancing which provides a better traffic distribution across ECMP paths or link-bundled paths in the core.

### Topology

This example illustrates two flows distributing over ECMPs and bundle links.



### Configure Load balancing using Control-Word and Flow-Label

Perform this task to configure load balancing using the **control-word** and **flow-label**.

```
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# pw-class path1
Router(config-l2vpn-pwc)# encapsulation mpls
Router(config-l2vpn-pwc)# control-word
Router(config-l2vpn-pwc-mpls)# load-balancing flow-label both
Router(config-l2vpn-pwc-mpls)# exit
Router(config-l2vpn-pwc)# exit
Router(config-l2vpn)# xconnect group grp1
Router(config-l2vpn-xc)# p2p vlan1
Router(config-l2vpn-xc-p2p)# interface HundredGigE0/0/0/1.2
Router(config-l2vpn-xc-p2p)# neighbor 10.0.0.2 pw-id 2000
Router(config-l2vpn-xc-p2p-pw)# pw-class path1
Router(config-l2vpn-xc-p2p-pw)# commit
```

### Running Configuration

This section shows the running configuration.

```
l2vpn
 pw-class path1
   encapsulation mpls
   control-word
   load-balancing
   flow-label both
 !
!
xconnect group grp1
 p2p vlan1
   interface HundredGigE0/0/0/1.2
   neighbor ipv4 10.0.0.2 pw-id 2000
   pw-class path1
 !
```

# L2VPN Traffic Load Balancing on PE Router

*Table 7: Feature History Table*

Feature Name	Release Information	Feature Description
Extended L2VPN Traffic Load Balancing Support for line cards and routers with the Q200 based Silicon One ASIC	Release 7.5.1	Introduced support for the following VLAN tags on line cards and routers with Q200 based Silicon One ASIC: <ul style="list-style-type: none"> <li>• Single VLAN tag 0x8100</li> <li>• QinQ outer 0x88A8 and inner 0x8100</li> </ul>
L2VPN Traffic Load Balancing on PE Router	Release 3.7.1	Distributes L2 VPN traffic across multiple physical links or paths.  Introduced support for the following VLAN tags on line cards and routers with Q100 based Silicon One ASIC: <ul style="list-style-type: none"> <li>• Single VLAN tag 0x8100</li> <li>• QinQ with outer 0x88A8 and inner 0x8100</li> </ul>

A Provider Edge (PE) router balances Layer 2 Virtual Private Network (L2VPN) traffic by efficiently distributing it across network paths using various load balancing techniques.

On Cisco 8000, different load balance methods are used for different traffic groups:

1. Traffic in VPWS (E-Line) Service - For more information, see [VPWS Service and Unicast Traffic in VPLS Service Load Balancing, on page 30](#)
2. Unicast Traffic in VPLS (E-LAN) Service - For more information, see [VPWS Service and Unicast Traffic in VPLS Service Load Balancing, on page 30](#).
3. Broadcast, Unknown Unicast, and Multicast (BUM) Traffic in VPLS (E-LAN) Service - For more information, see [BUM Traffic in VPLS Service Load Balancing, on page 30](#).

Load balance is orthogonal to point-to-point or multi-point connectivity. Load balance is needed over

1. Equal-Cost Multi-Path (ECMP) paths
2. Link Aggregation (LAG) bundle members

## Supported VLAN Tag Formats for Load Balancing

The Cisco 8000 load balances traffic when it is either not VLAN tagged or tagged with VLAN in supported formats.

Table 8: Supported VLAN Tag Formats

VLAN Format	Q100 supports VLAN from Release	Q200 supports VLAN from Release
Single VLAN Tag 0x8100	3.7.1	7.5.1
QinQ outer 0x88A8, inner 0x8100 (double VLAN tags)	3.7.1	7.5.1

## VPWS Service and Unicast Traffic in VPLS Service Load Balancing

The router performs load balancing on outgoing interfaces and bundle members in these scenarios:

- Ethernet frames entering from a L2 main or sub interface may be
  - switched out to another L2 interface that is part of a bundle, or
  - routed out to L3 interfaces with MPLS encapsulation.
- MPLS labeled PW and EVPN traffic entering from an L3 interface. After label disposition, the customer Ethernet frames may be
  - switched out to an L2 main or sub interface that is part of a bundle, or
  - routed out to L3 interfaces with MPLS encapsulation.

The router parses packets to identify the required headers for generating a load balance hash, which determines the path to route the traffic across the network. The hashing process varies based on whether the traffic is received on L2 or L3 interfaces.

For load balance hash on traffic received from L2 interfaces, refer to [Hash for Traffic Received on L2 Interface, on page 31](#).

For load balance hash on traffic received from L3 interfaces, refer to [Hash for Traffic Received on L3 Interface, on page 32](#).

## BUM Traffic in VPLS Service Load Balancing

The router performs BUM Traffic load balancing on outgoing interfaces and bundle members in these scenarios:

- Sending Traffic to an L2 Interface on bundle
- Sending Traffic over an MPLS PW
- Sending Traffic to EVPN MPLS Network

### Sending Traffic to an L2 Interface on Bundle

When sending traffic to a L2 interface, the router does not perform load balancing over bundle members. Instead, it pins all traffic sent to an L2 main or sub-interface to a single bundle member. The selection of the bundle member is based on the main or sub-interface ID.

### Sending Traffic over an MPLS PW

When BUM traffic is routed to an MPLS PW, the traffic is routed out to L3 interfaces. The router performs ECMP and bundle load balancing on the PW traffic. The hashing for load balancing is based on:

- PW VC label and flow label
- The 12 bytes of the PW payload. If control word (CW) is enabled, this includes a combination of 4 bytes of CW and 8 bytes of inner Ethernet MAC address.

#### **Sending Traffic to EVPN MPLS Network**

When BUM traffic is routed to an EVPN MPLS network, the traffic is encapsulated with:

- EVPN label
- ETREE leaf label or Ethernet Segment split horizon label

The MPLS encapsulated traffic is routed out to L3 interfaces. ECMP and bundle load balancing are performed on the EVPN MPLS encapsulated traffic. The hashing for load balancing is based on:

- EVPN label
- ETREE leaf label or Ethernet Segment split horizon label
- The 12 bytes of the PW payload. If CW is enabled, this includes a combination of 4 bytes of CW and 8 bytes of inner Ethernet MAC address.

## **MPLS non-IP Hash Disabled Configuration**

In certain configurations, you may choose to disable the non-IP hash mode. When non-IP hash mode is disabled, the Ethernet MAC address of BUM traffic isn't used to perform load balance hashing. This can impact how BUM traffic is distributed across the network, as the router relies on other available headers for hashing.

## **Hash for Traffic Received on L2 Interface**

For traffic received on a L2 interface, hashing depends on the headers present in the packet stack. The router generates the load balance hash using the designated fields based on the packet stack headers.

When ethernet frames entering a L2 main or sub interface, the router identifies the IP header within the packet based on the packet stack headers. The router uses these packet stack headers to generate the hash for traffic received on the L2 interface.

#### **IP Traffic on L2 Interface**

An ethernet frame is classified as IP traffic if it meets the following requirements:

- The ethernet header contains no more than two VLAN tags.
- The ethernet header either has no VLAN tag or has VLAN tags in a supported format as listed in [Supported VLAN Tag Formats for Load Balancing, on page 29](#).
- No more than ten MPLS labels are placed in front of the IP header.

#### **Non-IP Traffic on L2 Interface**

All traffic that does not meet the description of IP traffic on L2 Interface is classified as non-IP traffic on L2 interface.

### **L2 Hash Fields**

The L2 hash fields include the source and destination MAC addresses and the outer VLAN ID.

### **L3 Hash Fields**

The L3 hash fields include the source and destination IP addresses and the source and destination ports of the layer 4 header.

### **IP Traffic Load Balancing**

L2 hash fields were used in hashing. L3 hash fields were also included for limited traffic types.

### **Non-IP Traffic Load Balancing**

Non-IP traffic on the L2 interface is load balanced using L2 hash fields.

If any of the designated fields are missing, the router replaces those field values with zeros.

## **Hash for Traffic Received on L3 Interface**

For traffic received on a L3 interface hashing depends on several criteria, including the headers present in the packet stack and whether the Control Word (CW) and Flow Label (FL) are enabled on the pseudowire (PW).

When ethernet frames entering a L3 interface, the router identifies the IP header within the packet based on the packet stack headers.

### **IP over PW Traffic**

An ethernet frame is classified as IP over PW traffic if it meets the following criteria:

- The frame contains an outer ethernet header and an inner ethernet header.
- No more than two MPLS labels are placed between the outer ethernet header and the inner ethernet header.
- There is no control word in front of the inner ethernet header.
- The inner ethernet header contains no more than two VLAN tags.
- The inner ethernet header has no VLAN tag or has VLAN tags in supported format as listed in [Supported VLAN Tag Formats for Load Balancing, on page 29](#).
- Behind the inner ethernet header, there are no more than ten MPLS labels placed in front of the IP header.

### **Non-IP over PW Traffic**

All traffic that does not meet the IP over PW traffic criteria is classified as non-IP over PW traffic.

### **Inner Ethernet L2 Hash Fields**

The inner ethernet L2 hash fields include the source and destination MAC addresses, and the first VLAN tag of the inner ethernet frame.

### **Inner Ethernet L3 Hash Fields**

The inner ethernet L3 hash fields include L3 and L4 headers in the inner ethernet frame. They are source and destination IP addresses, the source and destination ports of the layer 4 header.



**Control Word Presence L2 Hash Fields**

The 12 bytes of the PW payload, which include 4 bytes of CW followed by the 8 bytes of inner ethernet frame MAC address.

**MPLS Hash Fields of Outer Ethernet Frame**

All MPLS labels in front of inner ethernet header.

**PW Disposition Traffic Load Balancing**

1. PW disposition traffic load balance when control word and flow label are both disabled.

**IP over PW Traffic Load Balancing**

IP over PW traffic load balance hash uses the inner ethernet L2 hash fields.

Inner ethernet L3 hash fields are also added in the hashing for limited types of traffic.

**Non-IP over PW Traffic Load Balancing**

Non-IP over PW traffic load balance hash always uses the inner ethernet L2 hash fields.

2. PW disposition traffic load balance when control word is enabled and flow label disabled.

In this case, all PW traffic is non-IP over PW traffic. Load balance hash uses control word presence L2 hash fields.

3. PW disposition traffic load balance when control word is disabled and flow label enabled.

IP over PW traffic load balancing hash uses the inner ethernet L2 hash fields. Inner ethernet L3 hash fields are also added in the hashing for limited types of traffic.

Non-IP over PW traffic load balancing hash uses the inner ethernet L2 hash fields.

4. PW disposition traffic load balance when control word and flow label are both enabled.

In this case, all PW traffic is Non-IP over PW traffic.

Non-IP over PW traffic load balance hash uses the control word presence L2 hash fields.

