



# Configuring Virtual Private LAN Service



## Note

The Cisco NX-OS release that is running on a managed device may not support all the features or settings described in this chapter. For the latest feature information and caveats, see the documentation and release notes for your platform and software release.

This chapter describes how to configure Virtual Private LAN Service (VPLS) using the Cisco Data Center Network Manager (DCNM) Access Circuits (ACs) for Layer 2 Virtual Private Networks (L2VPNs) on Cisco NX-OS devices.

This chapter includes the following sections:

- [Finding Feature Information, page 31-1](#)
- [Information About Virtual Private LAN Service, page 31-2](#)
- [Licensing Requirements for Virtual Private LAN Service, page 31-9](#)
- [Guidelines and Limitations for Virtual Private LAN Service, page 31-9](#)
- [Platform Support, page 31-11](#)
- [Configuring Access Circuits for Virtual Private LAN Service, page 31-11](#)
- [Verifying the Virtual Private LAN Service Configuration, page 31-31](#)
- [Monitoring Tunnel Interfaces, page 31-31](#)
- [Configuration Examples for Virtual Private LAN Service, page 31-31](#)
- [Field Descriptions for Tunnel Interfaces, page 31-10](#)
- [Additional References for Virtual Private LAN Service, page 31-35](#)
- [Feature History for Virtual Private LAN Service, page 31-37](#)

## Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at <https://tools.cisco.com/bugsearch/> and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the “New and Changed Information” chapter or the Feature History table below.

# Information About Virtual Private LAN Service

This section includes the following topics:

- [Layer 2 Services, page 31-2](#)
- [Attachment Circuits, page 31-2](#)
- [Pseudowire Interface, page 31-3](#)
- [Virtual Forwarding Interface, page 31-4](#)
- [Bridge Domain, page 31-4](#)
- [Ethernet Virtual Circuits, page 31-4](#)
- [Ethernet Flow Point, page 31-4](#)
- [Border Gateway Protocol Auto Discovery, page 31-5](#)
- [MAC Address Support, page 31-6](#)
- [Layer 2 VPN Stateful High Availability, page 31-7](#)
- [LinkSec, page 31-7](#)
- [MPLS Quality of Service, page 31-8](#)

## Layer 2 Services

A Layer 2 Virtual Private Network (L2VPN) enables service providers to carry multiple network services over a single converged network using Multiprotocol Label Switching (MPLS). MPLS L2VPN extends the Layer 2 domains in data centers. MPLS can be used to connect branch offices to back up data centers and also to interconnect multiple data centers in the same organization.

L2VPN services using the MPLS/IP core can be divided into two categories: wire and LAN services. The Virtual Private Wire Service (VPWS) provides point-to-point service between two customer edge (CE) devices over the provider core. The Virtual Private LAN Service (VPLS) provides point-to-multipoint service between multiple customer sites using a mesh of point-to-point pseudowires over the provider core to emulate a LAN between the sites.

## Attachment Circuits

A Layer 2 circuit that connects a customer edge (CE) node to a provider edge (PE) node is known as an attachment circuit or AC. A Layer 2 VPN (L2VPN) supports only Ethernet ACs on Cisco NX-OS devices.

To cross the network core, the Layer 2 traffic is tunneled inside a pseudowire. A pseudowire is typically a Multiprotocol Label Switching (MPLS) label-switched path (LSP), or a Layer 2 Tunneling Protocol (L2TP) tunnel, or the pseudowire can be locally switched from another AC. Layer 2 VPN connects different types of circuits (that is, different types of Layer 2 ACs and pseudowires) together in different ways to implement different types of end-to-end services.

The following types of ACs are supported:

- Ethernet port mode—This AC includes all frames that are sent and received on a physical Ethernet port.
- Ethernet 802.1Q—This AC includes all frames that are sent and received with a particular VLAN tag.

- Ethernet 802.1ad (Q-in-Q)—This AC includes all frames that are sent and received with a specific outer VLAN tag and a specific inner VLAN tag. VLAN-in-VLAN (Q-in-Q) is supported only in the service instance configuration and not in the subinterface configuration.
- Ethernet QinAny—This AC includes all frames that are sent and received with a specific outer VLAN tag and any inner VLAN tags, as long as the inner VLAN tag is not used on another subinterface.

An attachment circuit can participate in a Virtual Private LAN Service (VPLS) through a bridge domain. The Layer 2 switch port interfaces can also participate in VPLS forwarding. You can configure link bundles (port channels) with Ethernet Virtual Circuits (EVCs) to provide encapsulation types for link bundles.

## Pseudowire Interface

A pseudowire (PW) is a mechanism for emulating various networking or telecommunications services across packet-switched networks that use Ethernet, IP, or MPLS. A pseudowire interface (also known as a PW) in Cisco NX-OS is a logical interface type that represents a PW so that it can be consistently characterized in all communication and operations throughout the system.

You can create a static PW or dynamic PW configuration in pseudowire interface mode. Long form pseudowire interfaces must be explicitly configured using the appropriate Cisco NX-OS commands. Short-term, also known as auto-generated or dynamic, PWs are programmatically created and destroyed; you cannot configure a short-term PW. PW configurations can also be imported using a port profile.

With VPLS, different sites can share an Ethernet broadcast domain via PWs, providing any-to-any connectivity. VPLS uses a full mesh of Ethernet PWs to emulate a LAN segment or broadcast domain that is capable of learning and forwarding, based on Ethernet MAC addresses. The PW if-index is used as a handle for identification throughout the system; MAC entries are also acquired against these PWs.

## Control Word

According to RFC 4448, if a pseudowire (PW) is sensitive to packet misordering and is being carried over an MPLS packet switched network (PSN) that uses the contents of the MPLS payload to select the Equal Cost Multipath (ECMP), the PW must employ a mechanism that prevents packet misordering. This is necessary because ECMP implementations may examine the first nibble after the MPLS label stack to determine whether the labeled packet is IP or not. If the source MAC address of an Ethernet frame carried over the pseudowire without a control word present begins with 0x4 or 0x6, it can be mistaken for an IPv4 or an IPv6 packet. Depending on the configuration and topology of the MPLS network, this can lead to a situation where all packets for a given PW do not follow the same path, increasing out-of-order frames on a given PW or causing Operations, Administration, and Maintenance (OAM) packets to follow a different path than the actual traffic.

The Control Word Support feature provides the ability to sequence individual frames on the pseudowire, avoid ECMP paths, and perform OAM mechanisms including Virtual Circuit Connectivity Verification (VCCV).

## Virtual Forwarding Interface

A virtual forwarding interface (VFI) defines the configuration and the membership of the core pseudowires in the VPLS. A VFI is a virtual Layer 2 bridge that connects attachment circuits (physical Ethernet ports, logical Ethernet ports, or PWs) from customer edge (CE) devices to virtual circuits (VCs). The VFI is allocated an interface type and index in the system and is used by L2VPN and other components as an identifier.

## Bridge Domain

A bridge domain is a generic object that represents a Layer 2 broadcast domain on a device. VPLS uses a bridge domain to define a point-to-multipoint layer 2 service.

Creating a bridge domain also creates the underlying VLAN, if it does not already exist. There is a one-to-one mapping of bridge domains to VLANs; bridge domain 100 maps to VLAN 100.

## Ethernet Virtual Circuits

An Ethernet Virtual Circuit (EVC) as defined by the Metro Ethernet Forum is a port-level point-to-point or multipoint-to-multipoint Layer 2 circuit. It is an end-to-end representation of a single instance of a Layer 2 service being offered by a provider to a customer.

## Ethernet Flow Point

An Ethernet Flow Point (EFP) is the instantiation of an EVC on a specific interface on a device. The EFP interface representation is similar to that of a subinterface that maintains the parent-child relationship with the port.

The EFP interface is a Layer 2 logical interface. Any Layer 2 feature, protocol, or application that functions on a switchport is equally applicable to an EFP, all though some constraints might apply. Similar to a physical port, the interface state machine and forwarding behavior for the EFP depends on the service to which it belongs.

An EFP interface, also known as a service instance, is implicitly created when you configure an Ethernet service instance on a port. An EFP can be configured under a physical or logical parent port. Each service instance has its own configuration submode. Different features that apply to the service instance can be configured in that submode.

Because a single parent port can support multiple service instances, several EFPs can be associated with the port, with each EFP as part of a different EVC. For this reason, whenever a service instance is configured on a port, the port is internally brought up in trunk mode.

**Note**

---

The EVC represents a bridge domain. An EFP is an instance of an Ethernet flow on a particular interface, that belongs to a bridge domain. The Ethernet flow, not the entire port, belongs to the bridge domain.

---

## Flow per EFP

EVCs can identify flows based on multiple criteria in the Layer 2 header. In Cisco NX-OS, the flow identification for devices with Enhanced Address Recognition Logic (Earl) 8 hardware is based on matching the VLAN tag of the incoming packet. If the incoming packet has multiple VLAN tags only, the outer tag is used for traffic mapping to EFP.

Encapsulation defines the matching criteria that maps a VLAN to the service instance. A single VLAN ID can be configured for an exact match of the outermost tag. Any VLAN ID that is not specifically configured on an EFP or subinterface is treated as if it is implicitly configured for default encapsulation. On a parent port, you can configure either a single default EFP or one or more EFPs with explicit encapsulation, but not both.

## Border Gateway Protocol Auto Discovery

Border Gateway Protocol Auto Discovery (BGP-AD) automatically detects when provider edge (PE) devices are added to or removed from the VPLS domain, eliminating the need to manually configure PWs. BGP-AD can use either BGP or Label Distribution Protocol (LDP) signaling to exchange label binding information for supporting forwarding in an MPLS network.

The BGP-based auto discovery mechanism distributes Layer 2 VPN (L2VPN) endpoint provisioning information. BGP uses a separate L2VPN Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 VFI is configured. Prefix and path information is stored in the L2VPN database, allowing BGP to make best-path decisions. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services.

The VPLS BGP Signaling feature enables you to use BGP as the control plane protocol for both auto discovery and signaling in accordance with RFC 4761. Internal BGP (iBGP) peers exchange L2VPN AFI/SAFI update messages with L2VPN information to perform both auto discovery and signaling. The BGP multiprotocol Network Layer Reachability Information (NLRI) consists of a Route Distinguisher (RD), VPLS Endpoint ID (VE ID), VE Block Offset (VBO), VE Block Size (VBS), and Label Base (LB). Each NLRI consists of block labels such as LB, LB+1, ..., LB+VBS-1. The NLRI is exchanged between BGP devices for BGP auto discovery with BGP signaling.

Label Distribution Protocol (LDP)-based signaling follows the procedures specified in RFC 4447, which states that one provider edge device (PE1) sends a Label Mapping message to another PE device (PE2) to establish an LDP session in one direction. If the message is processed successfully, and there is no LDP session for the pseudowire in the opposite (PE2-to-PE1) direction, then PE2 sends a Label Mapping message to PE1.

For PE1 to begin signaling PE2, PE1 must know the address of the remote PE2. This information can be configured at PE1, or it can be generated dynamically through an auto-discovery procedure. The egress PE (PE1), which has knowledge of the ingress PE, initiates the setup by sending a Label Mapping message to the ingress PE (PE2), the Label Mapping message contains the FEC Tag Limit Values (TLV).

When the PE2 receives a Label Mapping message, PE2 interprets the message as a request to set up a pseudowire whose endpoint, PE2 is the forwarder. A Virtual Circuit (VC) or a pseudowire label is used to process packets at each PE device. Each PE device must reserve a PW label (local label) and advertise it to the peer. The VC label bindings exchanged over the targeted LDP session use the Forwarding Equivalence Classes (FEC) element type 128 via the LDP downstream unsolicited mode. Only one targeted session is created for multiple VCs between the PEs. If there already is a targeted session between the PEs by another application, then that session will be used. LDP will use the FEC type 128 to determine that the message is for the AToM application. LDP FEC 129 is used with auto discovery.

**Note**

VPLS with LDP signaling and no auto discovery is the most widely deployed solution.

## MAC Address Support

Layer 2 VPN (L2VPN) MAC address support is enabled by default when you configure a VPLS.

## MAC Address Flooding

One of the attributes of an Ethernet service is that frames sent to broadcast addresses and to unknown destination MAC addresses are flooded to all ports. To achieve flooding within the service provider network, all unknown unicast, broadcast and multicast frames are flooded over the corresponding pseudowires (PWs) to all provider edge (PE) nodes participating in the VPLS, as well as to all attachment circuits (ACs).

Multicast frames are different and do not necessarily have to be sent to all VPN members. For simplicity, the default approach of broadcasting multicast frames is used. To forward a frame, a PE must be able to associate a destination MAC address with a PW. VPLS-capable PEs have the capability to dynamically learn MAC addresses on both ACs and PWs and to forward and replicate packets across both ACs and PWs.

The MAC address table contains a list of the known MAC addresses and their forwarding information. In a typical VPLS architecture, the MAC address table and its management are distributed, which means that a copy of the MAC address table is maintained on the route processor (RP) card and the line cards.

## MAC Address Forwarding

A MAC address table contains address information that the switch uses to forward traffic between ports. All MAC addresses in the address table are associated with one or more ports. The table also contains a list of all known MAC addresses and their forwarding information. To forward a frame, a provider edge (PE) device must associate a destination MAC address with a pseudowire or an attachment circuit. This type of association is provided through a static configuration on each PE device or through dynamic learning that is flooded on all bridge ports.

When Layer 2 frames are received, VPLS does a lookup of the destination MAC address to learn the source MAC address. If the destination MAC address is not present in the MAC address table, the Layer 2 frames are flooded on the VLAN on which these frames were received. Flooded frames are sent on all configured pseudowires.

When Layer 2 frames are received on a pseudowire, the source MAC address is learned from the MAC address table by using the pseudowire port identifier (`peer_id`). If the destination MAC address is not present in the MAC address table, the frames are flooded on Layer 2 ports. If the destination MAC 2 address is present in the MAC address table, the frames are forwarded to the Layer 2 port or to the destination peer.

## MAC Address Learning

When a Layer 2 frame arrives on a bridge port, such as a pseudowire or an attachment circuit, and the source MAC address is unknown to the receiving provider edge (PE) device, the source MAC address is associated with the pseudowire or the attachment circuit. Outbound frames to the MAC address are forwarded to the appropriate pseudowire or attachment circuit.

MAC address learning uses the MAC address information that is learned from the hardware forwarding path. The number of learned MAC addresses is limited through configurable per-port and per-bridge domain MAC address limits.

## MAC Address Learning Aging

A timer is associated with the MAC addresses available in the MAC table. When this timer expires, the MAC addresses become invalid and are removed from the table. The relevant MAC entries are repopulated. This event is called MAC address aging. Provider edge (PE) devices must learn remote MAC addresses and directly attached MAC addresses on customer facing ports. MAC address learning derives topology and forwarding information from packets that originate at customer sites.

## MAC Address Withdrawal

VPLS MAC address withdrawal provides faster convergence by removing (or unlearning) MAC addresses that have been dynamically learned. No configuration is needed for enabling MAC address withdrawal support. Provider edge (PE) devices learn the remote MAC addresses and directly attached MAC addresses on customer-facing ports by deriving the topology and forwarding information from packets that originate at customer sites.

## Layer 2 VPN Stateful High Availability

The L2VPN Stateful High Availability (HA) feature uses two supervisor modules to provide uninterrupted service during a system failure. This implementation is the same for both Ethernet over Multiprotocol Label Switching (EoMPLS) and Virtual Private LAN Service (VPLS). During a failure, when an active supervisor is down, the standby supervisor seamlessly takes over all operations without disruptions. The supervisor modules also use Nonstop Forwarding (NSF), Stateful Switchover (SSO), and Graceful Restart (GR) for Any Transport over MPLS (AToM) to recover from an interruption in the service.

Peer Label Switch Routers (LSRs) exchange label binding information in an Multiprotocol Label Switching (MPLS) network to support the forwarding process. The MPLS Label Distribution Protocol Graceful Restart feature provides a mechanism by which the forwarding state between LSRs can be maintained during interruptions such as SSO failover events and temporary loss of Label Distribution Protocol (LDP) communication between the LSRs to enable NSF for MPLS traffic.

To enable NSF for Any Transport over MPLS (AToM) traffic, the provider edge (PE) devices and the LDP peers involved in the SSO event must support GR. There is no specific configuration required for Layer 2 VPN stateful HA.

## LinkSec

The LinkSec feature provides security for data centers over pseudowires using point-to-point encryption. LinkSec supports IEEE 802.1AE link-layer cryptography that provides hop-by-hop security of data in the MAC layer. Link-layer cryptography helps to ensure end-to-end data privacy while enabling the insertion of security service devices along the encrypted path.

## Hop-by-Hop Encryption

In this type of deployment, data is encrypted on the egress interface of the device and decrypted on the ingress interface of the device. Data is encrypted while being transmitted on interfaces but decrypted inside devices. However, if LinkSec is unavailable on certain segments of the network, data is sent in decrypted state on these segments. The advantage of this type of deployment is that Layer 2 Virtual Private Network (L2VPN) or Multiprotocol Label Switching (MPLS) is not aware of the encryption.

Hop-by-hop encryption is the default mode of encryption in LinkSec.

## Encryption and Decryption at Customer Edge Devices

After Layer 2 Virtual Private Network (L2VPN) or Multiprotocol Label Switching (MPLS) has added its label information to the frame, LinkSec encrypts both the data packet and the VLAN tag. The VLAN tag is lost and LinkSec sends the entire package across the network as a payload. In this type of deployment, data is encrypted and decrypted at customer edge (CE) devices only.

To enable this deployment, you should configure the provider edge (PE) ports in the port mode of the L2VPN operation because the VLAN tag is lost during LinkSec encryption.

This method can also be deployed by configuring the PE ports as access switch ports and mapping the packets that enter the ingress PE1 interface to an access VLAN. The packets are then forwarded using Virtual Private Lan Service (VPLS) or Ethernet over Multiprotocol Label Switching (EoMPLS) if the egress PE1 interface is configured to be part of a bridge domain of the VLAN.

## MPLS Quality of Service

To maintain the quality of service (QoS) when a packet traverses both Layer 2 and Layer 3 domains, the type of service (ToS) and CoS values must be mapped to each other. CoS refers to three bits in either an Inter-Switch Link (ISL) header or an 802.1Q header that are used to indicate the priority of an Ethernet frame as it passes through a switched network.

The 802.1Q provides QoS-based matching and marking to VLAN user priority bits to provide QoS on the Gigabit Ethernet WAN interface for 802.1Q packets. Packet marking helps identify packet flows. Packet marking enables the partitioning of a network into multiple priority levels or CoS. During network congestion, packets that are marked as priority are offered a higher priority than other packets.

802.1Q input packets are classified at eight different QoS levels (0 to 7) based on the VLAN user priority bits. For 802.1Q output packets, QoS marking is done at the VLAN header to modify VLAN user priority bits. QoS services use these priority bit settings to gain traffic priority during network congestion.

## Experimental Bits

EXP is a 3-bit field and part of a Multiprotocol Label Switching (MPLS) header. Experimental (EXP) bits in an MPLS header carry the priority of packets. Each label switching device along the network path honors the packet priority by queuing packets in the proper queue and servicing packets according to the priority. EXP bits define the quality of service (QoS) treatment (per-hop behavior) that a node should give to a packet. It is the equivalent of the differentiated service code point (DSCP) in the IP network. A DSCP defines a class and drop precedence. The EXP bits generally carry all information encoded in IP DSCP. However, in some cases, the EXP bits are used exclusively to encode the dropping precedence.

QoS on a Layer 2 VPN (L2VPN) network usually has two parts, an attachment circuit (AC) side and a pseudowire side. Layer 2 QoS is applied on the AC side and Layer 3 MPLS or IP QoS is applied on the pseudowire side.



Virtual Private LAN Service (VPLS) QoS is similar to Ethernet over MPLS (EoMPLS) QoS, except that QoS in VPLS is applied at ACs that participate in a VPLS bridge domain.

The core-facing MPLS interface must support a QoS policy. This QoS policy is applied on Ethernet Virtual Circuits (EVCs) and switchport interfaces. If a switchport interface participates in QoS handling, the matching criteria must include the VLAN on which VPLS forwarding is configured.

Setting the EXP bit value helps service providers who do not want to modify the value of the IP precedence field within the IP packets that are transported through their networks. By choosing different values for the Multiprotocol Label Switching (MPLS) EXP bit field, you can specify the priority that a packet requires during periods of network congestion. By default, the IP precedence value is copied into the MPLS EXP field during imposition. On the imposition path, packets are received from the AC and are sent toward the MPLS core. You can specify the MPLS EXP bits with an MPLS quality of service (QoS) policy.

By default, EXP is derived from COS for VPLS and VLAN-based EoMPLS. For port-based EoMPLS, by default, EXP is derived from the DSCP value.

## Licensing Requirements for Virtual Private LAN Service

The following table shows the licensing requirements for this feature:

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

## Guidelines and Limitations for Virtual Private LAN Service

Virtual Private LAN Service (VPLS) has the following configuration guidelines and limitations:

- Fabric Extender (FEX) ports are not supported as members of either XConnect or virtual forwarding instance (VFI) contexts.
- EoMPLS and VPLS can coexist on the same device.
- Ethernet over MPLS (EoMPLS) and VPLS can coexist with MPLS Layer 3 VPNs on the same device.
- VPLS and Cisco Overlay Transport Virtualization (OTV) can coexist in the same device if they are configured on different bridge domains or VLANs. A typical use case for this type of interaction involves a scenario where one cloud of the network uses OTV and the other cloud functions on an MPLS network using VPLS. A gateway facilitates data and packet forwarding between the two clouds. The OTV cloud and the MPLS cloud can be on the same physical network.
- The load balancing method required in the Layer 2 VPN is different from the Layer 3 VPN. Layer 3 VPN and Layer 2 VPN forwarding is performed independently on the device using two different types of adjacencies; therefore, the forwarding is not impacted by having a different method of load balancing for the Layer 2 VPN.

Ethernet virtual circuits (EVCs) have the following configuration guidelines and limitations:

- Ethernet flow points (EFPs) can be created only on Layer 3 interfaces without a switchport or IP address configuration.

- EFPs are not supported on subinterfaces.
- The total number of EFPs and subinterfaces that are supported in a system is 4000.
- The following features are not supported:
  - Service instance (EFP) group support.
  - EVC cross-connect and connect forwarding services.
  - Ethernet service protection features such as Ethernet Operations, Administration, and Maintenance (EOAM), Connectivity Fault Management (CFM), or Ethernet Local Management Interface (E-LMI).
  - Access control lists (ACLs).

Pseudowires have the following configuration guidelines and limitations:

- The maximum transmission unit (MTU) value of all pseudowires in a service must be the same. A pseudowire with an MTU value that differs from the MTU value of its peers will remain in a down state.
- Multicast and broadcast counters are not supported for pseudowires. All packets and bytes will be counted as unicast.

BGP-based auto discovery has the following configuration guidelines and limitations:

- BGP-based Virtual Private LAN Service (VPLS) auto discovery supports only IPv4 addresses.
- Auto discovery uses Forwarding Equivalence Class (FEC) 129 to convey endpoint information; manually configured pseudowires use FEC 128.
- Auto discovery is not supported with interautonomous system configurations.

## Field Descriptions for Tunnel Interfaces

This section includes the following field descriptions for tunnel interfaces:

- [Tunnel: Details Tab: Tunnel Details Section, page 31-10](#)
- [Tunnels: Details Tab: Source Section, page 31-11](#)
- [Tunnel: Statistics Tab, page 31-11](#)

### Tunnel: Details Tab: Tunnel Details Section

**Table 31-1** Tunnel: Details: Tunnel

Field	Description
Device	<i>Display only.</i> Name of device where tunnel interface exists.
Tunnel ID	<i>Display only.</i> Tunnel interface number.
Description	String that describes the tunnel interface.
Admin Status	Administrative status of the tunnel interface. The default is down.
Oper Status	Operational status of the tunnel interface.
MTU	MTU value for this tunnel.
IP Address	IPv4 address in dotted decimal notation.

**Table 31-1** Tunnel: Details: Tunnel

Field	Description
Net mask	Network mask for the IPv4 address, in dotted decimal notation.
IPv6 Address	IPv6 prefix in x:x:x::x/length format.

## Tunnels: Details Tab: Source Section

**Table 31-2** Tunnels: Details: Source

Field	Description
<b>Local Endpoint</b>	
Interface	Interface for the tunnel source address.
IP Address	IPv4 address, in dotted decimal notation for the tunnel source address.
<b>Remote Endpoint</b>	
Host Name	Device name for tunnel destination.
IP Address	IPv4 address, in dotted decimal notation for the tunnel destination address.

## Tunnel: Statistics Tab

**Table 31-3** Tunnel: Statistics Tab

Field	Description
Status	Status of statistics collection. Roll over Status to get a popup tip.
Select Parameters	List of statistics that can be gathered on tunnel interfaces.
Show Overview Chart	Overview popup of statistics.

## Platform Support

The following platform supports this feature. For platform-specific information, including guidelines and limitations, system defaults, and configuration limits, see the corresponding documentation.

Platform	Documentation
Cisco Nexus 7000 Series Switches	<a href="#">Cisco Nexus 7000 Series Switches Documentation</a>

## Configuring Access Circuits for Virtual Private LAN Service

This section contains the following topics:

- [Configuring an Ethernet Virtual Circuit for an 802.1Q Access Circuit, page 31-12](#)
- [Manually Configuring a Pseudowire Interface, page 31-15](#)
- [Configuring a Virtual Forwarding Interface for Static Pseudowires, page 31-17](#)

- [Configuring a Virtual Forwarding Interface for Auto Discovery](#), page 31-18
- [Customizing BGP-Based Auto Discovery Settings \(optional\)](#), page 31-24
- [Configuring Virtual Private LAN Service with a Bridge Domain](#), page 31-26
- [Configuring Virtual Private LAN Service with a VLAN](#), page 31-29

**Note**

If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

## Configuring an Ethernet Virtual Circuit for an 802.1Q Access Circuit

Repeat this task for each Ethernet Virtual Circuit (EVC) and Ethernet Flow Point (EFP) that you want to configure.

### Restrictions

- You can configure either a single default EFP or one or more EFPs with dot1q encapsulation on a parent port, but not both. Do not configure the **encapsulation default** command under an EFP unless it is the only service instance configured on the parent port.
- A maximum of 16 rewrite operations are supported per parent port on Cisco Nexus devices.
- No two EFPs for a parent port can have the same rewrite configuration.

### SUMMARY STEPS

1. **configure terminal**
2. **feature evc**
3. **interface ethernet** *slot/port*  
or  
**interface port-channel** *port-channel-number*
4. **no ip address** *ip-address mask*
5. **[no] service instance** *service-instance-id* **ethernet**
6. (Optional) **description** *description*
7. **encapsulation** { **default** | **dot1q** *vlan-id* }
8. (Optional) **rewrite ingress tag push dot1q** *vlan-id* **symmetric**
9. (Optional) **rewrite ingress tag translate 1-to-1 dot1q** *vlan-id* **symmetric**
10. (Optional) **copy running-config start-up config**

## DETAILED STEPS

	Command	Purpose
Step 1	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	<b>feature evc</b>  <b>Example:</b> switch(config)# feature evc	Enables Ethernet virtual circuits on the device.
Step 3	<b>interface ethernet</b> <i>slot/port</i> or <b>interface port-channel</b> <i>port-channel-number</i>  <b>Example:</b> switch(config)# interface ethernet 2/1 switch(config-if)# or switch(config)# interface port-channel 1 switch(config-if)#	Enters interface configuration mode and configures an interface.
Step 4	<b>no ip address</b> <i>ip-address mask</i>  <b>Example:</b> switch(config-if)# no ip address 10.1.1.1 255.255.255.0	Disables IP processing on an interface.
Step 5	[no] <b>service instance</b> <i>service-instance-id</i> <b>ethernet</b>  <b>Example:</b> switch(config-if)# service instance 1 ethernet switch(config-if-srv)#	Enters interface services configuration mode and configures an EFP on the interface. <ul style="list-style-type: none"> <li>The <i>service-instance-id</i> argument is a unique per-interface identifier for this EFP. The valid range is from 1 to 4000. The range might be restricted due to resource constraints.</li> </ul> <p><b>Note</b> You can use the <b>no</b> form of this command to delete the EFP and the associated configuration.</p>
Step 6	<b>description</b> <i>description</i>  <b>Example:</b> switch(config-if-srv)# description EFP1forVPLS	(Optional) Adds a description to this service instance configuration. <ul style="list-style-type: none"> <li>The maximum range for the <i>description</i> argument is 80 alphanumeric, case-sensitive characters.</li> </ul>

	Command	Purpose
Step 7	<p><b>encapsulation</b> {<b>default</b>   <b>dot1q</b> <i>vlan-id</i>}</p> <p><b>Example:</b>  <pre>switch(config-if-srv)# encapsulation default or switch(config-if-srv)# encapsulation dot1q 10</pre></p>	<p>Specifies that all dot1q frames that are otherwise unmatched by any other EFP are matched to this EFP.</p> <p><b>Note</b> You can enter the <b>encapsulation default</b> command only once in a parent port configuration.</p> <p>or</p> <p>Configures the matching criteria for mapping dot1q frames on an ingress interface to this EFP.</p> <ul style="list-style-type: none"> <li>The VLAN ID must match the domain ID of the bridge domain to which this EFP is to be associated. The valid range for the <i>vlan-id</i> argument is from 2 to 967.</li> </ul>
Step 8	<p><b>rewrite ingress tag push dot1q</b> <i>vlan-id</i> <b>symmetric</b></p> <p><b>Example:</b>  <pre>switch(config-if-srv)# rewrite ingress tag push dot1q 30 symmetric</pre></p>	<p>(Optional) Adds one VLAN tag to the incoming dot1q frame and symmetrically applies the operation to the ingress and egress frames.</p> <ul style="list-style-type: none"> <li>The VLAN ID must match the domain ID of the bridge domain to which this EFP is to be associated. The valid range for the <i>vlan-id</i> argument is from 2 to 967.</li> </ul> <p><b>Note</b> This command is supported only on an EFP configured with the <b>encapsulation default</b> command.</p>
Step 9	<p><b>rewrite ingress tag translate 1-to-1</b> <b>dot1q</b> <i>vlan-id</i> <b>symmetric</b></p> <p><b>Example:</b>  <pre>switch(config-if-srv)# rewrite ingress tag translate 1-to-1 dot1q 20 symmetric</pre></p>	<p>(Optional) Rewrites one VLAN tag in the incoming dot1q frame and symmetrically applies the operation to the ingress and egress frames.</p> <ul style="list-style-type: none"> <li>The VLAN ID must match the domain ID of the bridge domain to which this EFP is to be associated. The valid range for the <i>vlan-id</i> argument is from 2 to 967.</li> </ul> <p><b>Note</b> This command is supported only on an EFP configured with the <b>encapsulation dot1q</b> command.</p>
Step 10	<p><b>copy running-config startup-config</b></p> <p><b>Example:</b>  <pre>switch(config-if-srv)# copy running-config startup-config</pre></p>	<p>(Optional) Saves this configuration change.</p>

## What to Do Next

To bind this interface to a bridge domain, see the [“Configuring Virtual Private LAN Service with a Bridge Domain”](#) section.

## Manually Configuring a Pseudowire Interface

You can manually configure PWs for Access Circuits (ACs) or you can use BGP auto discovery (BGP-AD) to automatically generate PWs for the VPLS domain. To configure BGP-AD, see the “Configuring a Virtual Forwarding Interface for Auto Discovery” section.

### RESTRICTIONS

- If you manually configure multiple pseudowires and target different IP addresses on the same PE device for each pseudowire, do not use the same virtual circuit identifier (VC ID) to identify the pseudowires terminated at the same PE router.
- You cannot configure a pseudowire by manually configuring a neighbor on one PE device and using auto discovery on the other PE device to configure the same pseudowire in the other direction.

### SUMMARY STEPS

1. **configure terminal**
2. **[no] interface pseudowire *pw-id***
3. (Optional) **control word {exclude | include}**
4. (Optional) **description**
5. **mtu *size***
6. **neighbor *peer-ip-address vc-id***
7. **encapsulation mpls**
8. (Optional) **copy running-config start-up config**

### DETAILED STEPS

	Command	Purpose
Step 1	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	<b>[no] interface pseudowire <i>pw-id</i></b>  <b>Example:</b> switch(config)# interface pseudowire 12 switch(config-if-pseudowire)#	Enters interface pseudowire configuration mode and configures a static pseudowire logical interface. <ul style="list-style-type: none"> <li>• The <i>pw-id</i> argument is a unique per-interface identifier for this pseudowire. The range is from 1 to 200000. The range for a static pseudowire is from 1 to 8192.</li> </ul> <b>Note</b> You can use the <b>no</b> form of this command to delete the pseudowire interface and the associated configuration.

	Command	Purpose
Step 3	<p><b>control-word</b> {<b>exclude</b>   <b>include</b>}</p> <p><b>Example:</b>  <pre>switch(config-if-pseudowire)# control-word include</pre></p>	<p>(Optional) Enables control-word support.</p> <ul style="list-style-type: none"> <li>The <b>include</b> or <b>exclude</b> keywords specify whether the control word will or will not be included in the pseudowire packet.</li> <li>If you do not enable control word support in the pseudowire configuration, the default is autosense.</li> </ul> <p><b>Note</b> A device can receive a packet with or without the control word and the control word capability is negotiated with the peer. However, the device will not be able to generate a sequence number in the control word if the control word is added to the ingress device.</p>
Step 4	<p><b>description</b> <i>description</i></p> <p><b>Example:</b>  <pre>switch(config-if-pseudowire)# description longform</pre></p>	<p>(Optional) Adds a description to the interface configuration.</p> <ul style="list-style-type: none"> <li>The maximum range for the <i>description</i> argument is 254 alphanumeric, case-sensitive characters.</li> </ul>
Step 5	<p><b>mtu</b> <i>size</i></p> <p><b>Example:</b>  <pre>switch(config-if-pseudowire)# mtu 1400</pre></p>	<p>(Optional) Configures the maximum transmission unit (MTU) size, in bytes, for this interface.</p> <ul style="list-style-type: none"> <li>The valid range for the <i>size</i> argument is 576 to 9216. The default is 1500.</li> </ul>
Step 6	<p><b>neighbor</b> <i>peer-ip-address vc-id</i></p> <p><b>Example:</b>  <pre>switch(config-if-pseudowire)# neighbor 10.2.2.2 100</pre></p>	<p>Configures an emulated virtual circuit for this interface.</p> <ul style="list-style-type: none"> <li>The combination of the <i>peer-ip-address</i> and <i>vc-id</i> arguments must be unique on a device.</li> <li>The peer IP address is the address of the provider edge (PE) peer.</li> <li>The <i>vc-id</i> argument is an identifier for the virtual circuit between devices. The valid range is from 1 to 4294967295.</li> </ul>
Step 7	<p><b>encapsulation</b> <b>mpls</b></p> <p><b>Example:</b>  <pre>switch(config-if-pseudowire)# encapsulation mpls switch(config-pseudowire-mpls)#</pre></p>	<p>Enters pseudowire MPLS configuration mode and specifies MPLS encapsulation for this interface.</p>
Step 8	<p><b>copy running-config startup-config</b></p> <p><b>Example:</b>  <pre>switch(config-pseudowire-mpls)# copy running-config startup-config</pre></p>	<p>(Optional) Saves this configuration change.</p>



# Configuring a Virtual Forwarding Interface for Static Pseudowires

## BEFORE YOU BEGIN

Ensure that you have configured the PWs.

## RESTRICTIONS

- You can configure both auto discovered and manually configured pseudowires in a single virtual forwarding instance (VFI). However, the pseudowires cannot go to the same peer PE device.
- You cannot configure a pseudowire by manually configuring a neighbor on one PE device and using auto discovery on the other PE device to configure the same pseudowire in the other direction.

## SUMMARY STEPS

1. `configure terminal`
2. `[no] l2vpn vfi context vfi-name`
3. (Optional) `description description`
4. `vpn vpn-id`
5. `member pseudowire pw-id`
6. (Optional) `copy running-config start-up config`

## DETAILED STEPS

	Command	Purpose
Step 1	<code>configure terminal</code>  <b>Example:</b> switch# <code>configure terminal</code> switch(config)#	Enters global configuration mode.
Step 2	<code>[no] l2vpn vfi context vfi-name</code>  <b>Example:</b> switch(config)# <code>l2vpn vfi context foo</code> switch(config-l2vpn-vfi)#	Establishes a Layer 2 VPN (L2VPN) Virtual Forwarding Interface (VFI) between two or more separate networks. <ul style="list-style-type: none"> <li>• The <i>vfi-name</i> argument is a unique per-interface identifier for this VFI. The maximum range is 100 alphanumeric, case-sensitive characters.</li> </ul> <b>Note</b> You can use the <b>no</b> form of this command to delete the VFI and the associated configuration.
Step 3	<code>description description</code>  <b>Example:</b> switch(config-l2vpn-vfi)# <code>description PWsforVPLS</code>	(Optional) Adds a description to the interface configuration. <ul style="list-style-type: none"> <li>• The maximum range for the <i>description</i> argument is 254 alphanumeric characters.</li> </ul>

	Command	Purpose
Step 4	<b>vpn</b> <i>vpn-id</i>  <b>Example:</b> switch(config-l2vpn-vfi)# vpn 100	Configures a Virtual Private Network (VPN) ID on a VFI context. <ul style="list-style-type: none"> <li>The valid range is from 1 to 4294967295.</li> </ul>
Step 5	<b>member pseudowire</b> <i>pw-id</i>  <b>Example:</b> switch(config-l2vpn-vfi)# member pseudowire 12	Binds a static pseudowire to this VFI. <ul style="list-style-type: none"> <li>This command is supported for a static pseudowire only.</li> <li>The <i>pw-id</i> argument is a unique per-interface identifier for a static pseudowire. The valid range is from 1 to 8192.</li> <li>Repeat this step for each static pseudowire to be associated with this VFI.</li> </ul>
Step 6	<b>copy running-config startup-config</b>  <b>Example:</b> switch(config-l2vpn-vfi)# copy running-config startup-config	(Optional) Saves this configuration change.

## Configuring a Virtual Forwarding Interface for Auto Discovery

Perform just one of the following tasks:

- [Configuring BGP Auto Discovery and BGP Signaling, page 31-18](#)
- [Configuring BGP Auto Discovery and LDP Signaling, page 31-22](#)

## Configuring BGP Auto Discovery and BGP Signaling

### RESTRICTIONS

- You can configure both auto discovered and manually configured pseudowires in a single virtual forwarding instance (VFI). However, the pseudowires cannot go to the same peer PE device.
- You cannot configure a pseudowire by manually configuring a neighbor on one PE device and using auto discovery on the other PE device to configure the same pseudowire in the other direction.
- After enabling VPLS autodiscovery, if you manually configure a neighbor by using the member command and both peers are in autodiscovery mode, each peer will receive discovery data for that VPLS. To prevent peers from receiving data for the VPLS domain, manually configure route target (RT) values. For information, see the “[Customizing BGP-Based Auto Discovery Settings \(optional\)](#)” section.

### SUMMARY STEPS

1. **configure terminal**
2. **[no] l2vpn vfi context** *vfi-name*
3. (Optional) **description** *description*
4. **vpn** *vpn-id*

5. **autodiscovery bgp signaling bgp**
6. **ve id** *ve-id-number*
7. **ve range** *range*
8. **router bgp** *as-number*
9. **bgp graceful restart**
10. **neighbor** *peer-ip-address vc-id remote as as-number*
11. **address-family l2vpn vpls**
12. **neighbor** [*peer-ip-address | peer-group-name*] **activate**
13. **neighbor** [*peer-ip-address | peer-group-name*] **send-community extend**
14. **neighbor** [*peer-ip-address | peer-group-name*] **suppress-signaling-protocol ldp**
15. Repeat Steps 11 to 15 to configure additional neighbors in an L2VPN address family.
16. (Optional) **copy running-config start-up config**

## DETAILED STEPS

	Command	Purpose
Step 1	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] <b>l2vpn vfi context</b> <i>vfi-name</i>  <b>Example:</b> switch(config)# l2vpn vfi context foo switch(config-l2vpn-vfi)#	Establishes a Layer 2 VPN (L2VPN) Virtual Forwarding Interface (VFI) context between two or more separate networks. <ul style="list-style-type: none"> <li>The <i>vfi-name</i> argument is a unique per-interface identifier for this context. The maximum range is 100 alphanumeric, case-sensitive characters.</li> </ul> <b>Note</b> You can use the <b>no</b> form of this command to delete the context and the associated configuration.
Step 3	<b>description</b> <i>description</i>  <b>Example:</b> switch(config-l2vpn-vfi)# description PWsforVPLS	(Optional) Adds a description to the interface configuration. <ul style="list-style-type: none"> <li>The maximum range for the <i>description</i> argument is 254 alphanumeric characters.</li> </ul>
Step 4	<b>vpn</b> <i>vpn-id</i>  <b>Example:</b> switch(config-l2vpn-vfi)# mtu 1400	Configures a Virtual Private Network (VPN) ID on a VFI context. <ul style="list-style-type: none"> <li>The valid range is from 1 to 4294967295.</li> </ul>

	Command	Purpose
Step 5	<pre>autodiscovery bgp signaling bgp</pre> <p><b>Example:</b>  <pre>switch(config-l2vpn-vfi)# autodiscovery bgp signaling bgp</pre></p>	Enables BGP auto discovery and BGP signaling.
Step 6	<pre>ve id ve-id-number</pre> <p><b>Example:</b>  <pre>switch(config-l2vpn-vfi)# ve id 1</pre></p>	<p>Configures a VPLS Endpoint ID (VEID) for the NLRI exchanged between BGP devices.</p> <ul style="list-style-type: none"> <li>Repeat this step to add each additional VE ID. The VE ID must be unique within the same VPLS domain for all PE devices.</li> </ul> <p><b>Note</b> Numbering sequences such as 1,2,3 or 501, 502, 503 are good because the VEIDs are contiguous. A numbering scheme such as 100, 200, 300 is bad because it is noncontiguous.</p> <ul style="list-style-type: none"> <li>If you change the VEID, the virtual circuit (VC) reprovisions, and as a result, traffic is impacted.</li> </ul>
Step 7	<pre>ve range ve-range-number</pre> <p><b>Example:</b>  <pre>switch(config-l2vpn-vfi)# ve range</pre></p>	<p>(Optional) Configures the number of VEIDs for the Autonomous System (AS).</p> <ul style="list-style-type: none"> <li>The range for the <i>ve-range-number</i> argument is from 1 to 100. The default is 10.</li> <li>The VE range can be configured based on the number of neighboring PE devices in the network. The VE range value should be approximately the same as the number of neighbors (up to 100).</li> <li>If no VE range is configured or an existing VE range value is removed, then the default VE range is applied. The default VE range should not be used if the router has many PE neighbors.</li> <li>If you change the VE range, the virtual circuit (VC) reprovisions and as a result, traffic is impacted.</li> </ul>
Step 8	<pre>router bgp as-number</pre> <p><b>Example:</b>  <pre>switch(config-l2vpn-vfi)# router bgp 100 switch(config-router)#</pre></p>	<p>Enters the router BGP configuration mode and assigns an autonomous system (AS) number to the local BGP speaker device.</p> <ul style="list-style-type: none"> <li>The <i>as-number</i> argument identifies the device to other BGP devices and tags the routing information to be passed along. The range is from 1 to 65535.</li> <li>The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in the <i>xx.xx</i> format.</li> </ul>

	Command	Purpose
Step 9	<b>bgp graceful restart</b>  <b>Example:</b> switch(config-router)# bgp graceful restart	Enables the graceful restart and the graceful restart helper capability.
Step 10	<b>neighbor peer-ip-address remote-as as-number</b>  <b>Example:</b> switch(config-router)# neighbor 10.1.1.1 remote-as 100	<p>Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local device.</p> <ul style="list-style-type: none"> <li>• The combination of the <i>peer-ip-address</i> and <i>as-number</i> arguments must be unique on a device.</li> <li>• The peer IP address is the address of the provider edge (PE) peer.</li> <li>• If the <i>as-number</i> argument matches the autonomous system number specified in the <b>router bgp</b> command, the neighbor is an internal neighbor.</li> <li>• If the <i>as-number</i> argument does not match the autonomous system number specified in the <b>router bgp</b> command, the neighbor is an external neighbor.</li> </ul>
Step 11	<b>address-family l2vpn vpls</b>  <b>Example:</b> switch(config-router)# address-family l2vpn vpls switch(config-router-af)#	Creates an L2VPN address family session and specifies that VPLS endpoint provisioning information is to be distributed to BGP peers.
Step 12	<b>neighbor [peer-ip-address   peer-group-name] activate</b>  <b>Example:</b> switch(config-router-af)# neighbor 10.10.10.1 activate	Enables the exchange of information with the specified BGP neighbor
Step 13	<b>neighbor [peer-ip-address   peer-group-name] send-community extend</b>  <b>Example:</b> switch(config-router-af)# neighbor 10.10.10.1 send-community extend	Specifies that a community attribute should be sent to the BGP neighbor.
Step 14	<b>neighbor [peer-ip-address   peer-group-name] suppress-signaling-protocol ldp</b>  <b>Example:</b> switch(config-router-af)# neighbor 10.10.10.1 suppress-signaling-protocol ldp	Suppresses LDP signaling for a BGP neighbor so that BGP signaling for auto discovery is used.

	Command	Purpose
Step 15	Repeat Steps 11 to 15 to configure additional neighbors in an L2VPN address family.	
Step 16	<code>copy running-config startup-config</code>	(Optional) Saves this configuration change.
	<b>Example:</b> <pre>switch(config-router-af)# copy running-config startup-config</pre>	

## Configuring BGP Auto Discovery and LDP Signaling

### RESTRICTIONS

- You can configure both auto discovered and manually configured pseudowires in a single virtual forwarding instance (VFI). However, the pseudowires cannot go to the same peer PE device.
- You cannot configure a pseudowire by manually configuring a neighbor on one PE device and using auto discovery on the other PE device to configure the same pseudowire in the other direction.
- After enabling VPLS autodiscovery, if you manually configure a neighbor by using the member command and both peers are in autodiscovery mode, each peer will receive discovery data for that VPLS. To prevent peers from receiving data for the VPLS domain, manually configure route target (RT) values. For information, see the [“Customizing BGP-Based Auto Discovery Settings \(optional\)”](#) section.

### SUMMARY STEPS

1. configure terminal
2. `[no] l2vpn vfi context vfi-name`
3. (Optional) `description description`
4. `vpn vpn-id`
5. `autodiscovery bgp signaling ldp`
6. `router bgp as-number`
7. `neighbor peer-ip-address vc-id`
8. `address-family l2vpn vpls`
9. (Optional) `copy running-config start-up config`

## DETAILED STEPS

	Command	Purpose
Step 1	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] <b>l2vpn vfi context</b> <i>vfi-name</i>  <b>Example:</b> switch(config)# l2vpn vfi context foo switch(config-l2vpn-vfi)#	Establishes a Layer 2 VPN (L2VPN) Virtual Forwarding Interface (VFI) context between two or more separate networks. <ul style="list-style-type: none"> <li>The <i>vfi-name</i> argument is a unique per-interface identifier for this context. The maximum range is 100 alphanumeric, case-sensitive characters.</li> </ul> <b>Note</b> You can use the <b>no</b> form of this command to delete the context and the associated configuration.
Step 3	<b>description</b> <i>description</i>  <b>Example:</b> switch(config-l2vpn-vfi)# description PWsforVPLS	(Optional) Adds a description to the interface configuration. <ul style="list-style-type: none"> <li>The maximum range for the <i>description</i> argument is 254 alphanumeric characters.</li> </ul>
Step 4	<b>vpn</b> <i>vpn-id</i>  <b>Example:</b> switch(config-l2vpn-vfi)# mtu 1400	Configures a Virtual Private Network (VPN) ID on a VFI context. <ul style="list-style-type: none"> <li>The valid range is from 1 to 4294967295.</li> </ul>
Step 5	<b>autodiscovery bgp signaling ldp</b>  <b>Example:</b> switch(config-l2vpn-vfi)# autodiscovery bgp signaling ldp	Enables BGP auto discovery and LDP signaling.
Step 6	<b>router bgp</b> <i>as-number</i>  <b>Example:</b> switch(config-l2vpn-vfi)# router bgp 100 switch(config-router)#	Enters the router BGP configuration mode and assigns an autonomous system (AS) number to a device. <ul style="list-style-type: none"> <li>The <i>as-number</i> argument identifies the device to other BGP devices and tags the routing information to be passed along. the range is from 1 to 65535.</li> </ul>

	Command	Purpose
Step 7	<p><b>neighbor</b> <i>peer-ip-address</i> <b>remote-as</b> <i>as-number</i></p> <p><b>Example:</b>  <pre>switch(config-router)# neighbor 10.1.1.1 remote-as 100</pre></p>	<p>Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local device.</p> <ul style="list-style-type: none"> <li>The combination of the <i>peer-ip-address</i> and <i>as-number</i> arguments must be unique on a device.</li> <li>The peer IP address is the address of the provider edge (PE) peer.</li> <li>If the <i>as-number</i> argument matches the autonomous system number specified in the <b>router bgp</b> command, the neighbor is an internal neighbor.</li> <li>If the <i>as-number</i> argument does not match the autonomous system number specified in the <b>router bgp</b> command, the neighbor is an external neighbor.</li> </ul>
Step 8	<p><b>address-family</b> l2vpn vpls</p> <p><b>Example:</b>  <pre>switch(config-router)# address-family l2vpn vpls</pre></p>	<p>Creates an L2VPN address family session and specifies that VPLS endpoint provisioning information is to be distributed to BGP peers.</p>
Step 9	<p><b>copy running-config startup-config</b></p> <p><b>Example:</b>  <pre>switch(config-router)# copy running-config startup-config</pre></p>	<p>(Optional) Saves this configuration change.</p>

## Customizing BGP-Based Auto Discovery Settings (optional)

### Before You Begin

Ensure that you have configured BGP-based auto discovery for VPLS.

### SUMMARY STEPS

- configure terminal**
- [no] l2vpn vfi context** *vfi-name*
- (Optional) **vpls-id** { *autonomous-system-number:nn* | *ip-address:nn* }
- (Optional) **rd** { *autonomous-system-number:nn* | *ip-address:nn* }
- (Optional) **auto-route-target**  
or  
(Optional) **route-target** [**import** | **export** | **both**] { *autonomous-system-number:nn* | *ip-address:nn* }
- (Optional) **copy running-config startup-config**



## DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><b>configure terminal</b></p> <p><b>Example:</b>  switch# configure terminal  switch(config)#</p>	Enters global configuration mode.
Step 2	<p><b>[no] l2vpn vfi context</b>  <i>vfi-name</i></p> <p><b>Example:</b>  switch(config)# l2vpn vfi  context foo  switch(config-l2vpn-vfi)#</p>	<p>Establishes a Layer 2 VPN (L2VPN) Virtual Forwarding Interface (VFI) context between two or more separate networks.</p> <ul style="list-style-type: none"> <li>The <i>vfi-name</i> argument is a unique per-interface identifier for this context. The maximum range is 100 alphanumeric, case-sensitive characters.</li> </ul> <p><b>Note</b> You can use the <b>no</b> form of this command to delete the context and the associated configuration.</p>
Step 3	<p><b>vpls-id</b>  {<i>autonomous-system-number:nn</i>    <i>ip-address:nn</i>}</p> <p><b>Example:</b>  switch(config-l2vpn-vfi)#  vpls-id 5:200</p>	<p>(Optional) Changes the value of the VPLS ID from the generated value to the specified value.</p> <ul style="list-style-type: none"> <li>Auto discovery automatically generates a VPLS ID using the Border Gateway Protocol (BGP) autonomous system (AS) number and the configured Virtual Private Network (VPN) ID on the VFI context.</li> <li>The value for the <i>nn</i> argument is the network number.</li> </ul>
Step 4	<p><b>rd</b>  {<i>autonomous-system-number:nn</i>    <i>ip-address:nn</i>}</p> <p><b>Example:</b>  switch(config-l2vpn-vfi)# rd  2:2</p>	<p>(Optional) Changes the value of the route distinguisher (RD) from the generated value to the specified value.</p> <ul style="list-style-type: none"> <li>Auto discovery automatically generates an RD using the BGP autonomous system number (AS) and the configured Virtual Private Network (VPN) ID on the VFI context.</li> <li>The value for the <i>nn</i> argument is the network number. The network number must be preceded by a colon (:).</li> </ul>

	Command or Action	Purpose
Step 5	<pre> <b>auto-route-target</b> or <b>route-target</b> [<b>import</b>   <b>export</b>   <b>both</b>] { <i>autonomous-system-number:nn</i>   <i>ip-address:nn</i> }  <b>Example:</b> switch(config-l2vpn-vfi)# route-target 600:2222 </pre>	<p>(Optional) Enables auto discovery to generate a route target (RT) using the lower 6 bits of the RD and the configured Virtual Private Network (VPN) ID on the VFI context.</p> <ul style="list-style-type: none"> <li>This is the default. If you previously configured the <b>route-target</b> command, use this command to change the explicitly configured RT to a generated RT.</li> </ul> <p>or</p> <p>(Optional) Changes the value of the route target (RT) from the generated value to the specified value.</p> <ul style="list-style-type: none"> <li>The value for the <i>nn</i> argument is the network number. The network number must be preceded by a colon (:).</li> </ul>
Step 6	<pre> <b>copy running-config</b> <b>startup-config</b>  <b>Example:</b> switch(config-l2vpn-vfi)# copy running-config startup-config </pre>	<p>(Optional) Saves this configuration change.</p>

## Configuring Virtual Private LAN Service with a Bridge Domain

You can configure VPLS either with a bridge domain or with a VLAN. To associate a VFI directly to a VLAN, go to the [“Configuring Virtual Private LAN Service with a VLAN”](#) section on page 31-29.

### BEFORE YOU BEGIN

- Ensure that you have configured the VFI.
- Ensure that you have configured an EFP for the 802.1Q Access Circuit (AC).

### Restrictions

Switchport VLANs and EFPs cannot be associated with the same bridge domain.

### SUMMARY STEPS

- configure terminal**
- feature mpls l2vpn**
- feature evc**
- system bridge-domain** *id* [*-id* | *-id,...,id-id*]
- interface ethernet** *slot/port*  
or  
**interface port-channel** *port-channel-number*
- [**no**] **service instance** *service-instance-id* **ethernet**
- (Optional) **description** *description*
- encapsulation dot1q** *vlan-id*

9. **[no] bridge-domain** *domain-id*
10. **member vfi** *vfi-id*
11. **member interface slot/port service instance** *service-instance-id*
12. (Optional) **copy running-config start-up config**

## DETAILED STEPS

	Command	Purpose
Step 1	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	<b>[no] feature mpls l2vpn</b>  <b>Example:</b> switch(config)# feature mpls l2vpn	Enables Multiprotocol Label Switching (MPLS) Layer 2 VPN (L2VPN) features.  <b>Note</b> Using the no feature <b>mpls l2vpn</b> command removes all existing L2VPN configurations. Using the feature <b>mpls l2vpn</b> command again does not restore the earlier L2VPN configuration.
Step 3	<b>feature evc</b>  <b>Example:</b> switch(config)# feature evc	Enables Ethernet virtual circuits on the device.
Step 4	<b>system bridge-domain id [-id   -id, ..., id-id]</b>  <b>Example:</b> switch(config)# system bridge-domain 10-50,100-500	Identifies the IDs that are available for bridge-domain configurations. <ul style="list-style-type: none"> <li>• The valid range for the <i>id</i> argument is from 2 to 967.</li> <li>• The optional <i>-id</i> keyword and argument combination identifies the last ID in a range of contiguous IDs. The hyphen (-) is required.</li> <li>• The optional list of ID ranges are separated by commas (.). Do not type the ellipses (...).</li> </ul>
Step 5	<b>interface ethernet slot/port</b> or <b>interface port-channel</b> <i>port-channel-number</i>  <b>Example:</b> switch(config)# interface ethernet 2/1 switch(config-if)# or switch(config)# interface port-channel 1 switch(config-if)#	Enters interface configuration mode.

	Command	Purpose
Step 6	<p>[no] <b>service instance</b> <i>service-instance-id</i> <b>ethernet</b></p> <p><b>Example:</b>  <pre>switch(config-if)# service instance 1 ethernet switch(config-if-srv)#</pre></p>	<p>Enters interface services configuration mode and configures an EFP on the interface.</p> <ul style="list-style-type: none"> <li>The <i>service-instance-id</i> argument is a unique per-interface identifier for this EFP. The valid range is from 1 to 4000. The range might be restricted due to resource constraints.</li> </ul> <p><b>Note</b> You can use the <b>no</b> form of this command to delete the EFP and the associated configuration.</p>
Step 7	<p><b>description</b> <i>description</i></p> <p><b>Example:</b>  <pre>switch(config-if-srv)# description EFP1forVPLS</pre></p>	<p>(Optional) Adds a description to this service instance configuration.</p> <ul style="list-style-type: none"> <li>The maximum range for the <i>description</i> argument is 80 alphanumeric, case-sensitive characters.</li> </ul>
Step 8	<p><b>encapsulation dot1q</b> <i>vlan-id</i></p> <p><b>Example:</b>  <pre>switch(config-if-srv)# encapsulation dot1q 100</pre></p>	<p>Allows flow from the specified VLAN ID to pass through the EFP.</p> <ul style="list-style-type: none"> <li>The VLAN ID must match the domain ID of the bridge domain to which this EFP is to be associated. The valid range for the <i>vlan-id</i> argument is from 2 to 967.</li> </ul>
Step 9	<p>[no] <b>bridge-domain</b> <i>domain-id</i></p> <p><b>Example:</b>  <pre>switch(config-if-srv)# bridge-domain 100 switch(config-bdomain)#</pre></p>	<p>Enters bridge-domain configuration mode and configures a bridge domain.</p> <ul style="list-style-type: none"> <li>The <i>domain-id</i> argument is a unique identifier for the bridge domain and underlying VLAN to be created. The valid range is defined by the system bridge-domain configuration.</li> </ul> <p><b>Note</b> You can use the <b>no</b> form of this command to remove the bridge-domain configuration including port associations. Removing the bridge-domain configuration does not remove the underlying VLAN. If a VLAN is associated with a bridge domain, you cannot remove the VLAN without first removing the bridge domain. To remove the underlying VLAN, use the <b>no vlan</b> command after you remove the bridge domain.</p>
Step 10	<p><b>member vfi</b> <i>vfi-id</i></p> <p><b>Example:</b>  <pre>switch(config-bdomain)# member vfi foo</pre></p>	<p>(Optional) Binds a VFI to this bridge domain.</p> <ul style="list-style-type: none"> <li>The <i>vfi-id</i> argument identifies the VFI to be bound. The maximum range is 100 alphanumeric, case-sensitive characters.</li> </ul>

	Command	Purpose
Step 11	<pre>member interface slot/port service instance service-instance-id</pre> <p><b>Example:</b></p> <pre>switch(config-bdomain)# member ethernet 2/1 service instance 1</pre>	(Optional) Binds a service instance to this bridge domain. <ul style="list-style-type: none"> <li>• The <i>interface slot/port</i> argument identifies the interface under which the service instance is configured.</li> <li>• The <i>service-instance-id</i> argument identifies the service instance to be bound. The valid range is from 1 to 4000.</li> </ul>
Step 12	<pre>copy running-config startup-config</pre> <p><b>Example:</b></p> <pre>switch(config-bdomain)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

## Configuring Virtual Private LAN Service with a VLAN

You can configure VPLS either with a bridge domain or with a VLAN. To associate the VFI (or EFP) to a bridge domain, see the [“Configuring Virtual Private LAN Service with a Bridge Domain”](#) section on page 31-26.

### BEFORE YOU BEGIN

Ensure that you have configured the VFI.

### SUMMARY STEPS


1. **configure terminal**
2. **[no] vlan vlan-id**
3. **member vfi vfi-id**
4. **exit**
5. **interface ethernet slot/port**
6. **switchport mode trunk**
7. **switchport allowed vlan vlan-id**
8. (Optional) **copy running-config start-up config**

## DETAILED STEPS

	Command	Purpose
Step 1	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] <b>vlan</b> <i>domain-id</i>  <b>Example:</b> switch(config)# vlan 100 switch(config-vlan)#	Enters VLAN configuration mode and configures a VLAN.  <ul style="list-style-type: none"> <li>The <i>vlan-id</i> argument is a unique identifier for the VLAN. The valid range is from 1 to 4094.</li> </ul> <b>Note</b> You can use the <b>no</b> form of this command to remove the VLAN configuration including port associations.
Step 3	<b>member vfi</b> <i>vfi-id</i>  <b>Example:</b> switch(config-vlan)# member vfi foo	Binds a VFI to this VLAN.  <ul style="list-style-type: none"> <li>The <i>vfi-id</i> argument identifies the VFI to be bound. The maximum range is 100 alphanumeric, case-sensitive characters.</li> </ul>
Step 4	<b>exit</b>  <b>Example:</b> switch(config-vlan)# exit switch (config)#	Exits VLAN configuration mode.
Step 5	<b>interface ethernet</b> <i>slot/port</i>  <b>Example:</b> switch(config)# interface ethernet 2/1 switch(config-if)#	Enters interface configuration mode and configures an Ethernet interface.
Step 6	<b>switchport mode trunk</b>  <b>Example:</b> switch(config-if)# switchport mode trunk	Sets the interface type to be a Layer 2 host port for a trunk.
Step 7	<b>switchport allowed vlan</b> <i>vlan-id</i>  <b>Example:</b> switch(config-if)# switchport allowed vlan 100	Allows flow from the specified VLAN to pass through the trunk.  <ul style="list-style-type: none"> <li>The VLAN ID must match the ID of the VLAN to which this VFI is to be associated. The valid range for the <i>vlan-id</i> argument is from 1 to 4094.</li> </ul>
Step 8	<b>copy running-config startup-config</b>  <b>Example:</b> switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

## Verifying the Virtual Private LAN Service Configuration

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

Command	Purpose
<code>show ethernet service instance [detail]</code>	Displays information about service instances that are configured on the device.
<code>show ethernet service instance interface ethernet slot/port [detail]</code>	Displays information about service instances that are configured on an interface.
<code>show interface [brief description]</code>	Displays the interface status and information.
<code>show interface pseudowire pw-id</code>	Displays the status and information about the specified interface.
<code>show interface pseudowire pw-id brief</code>	Displays brief information about the specified interface.
<code>show interface pseudowire pw-id counters</code>	Displays the in and out counters for the specified interface.
	 <p><b>Note</b> Multicast and broadcast counters are not supported for pseudowires. All packets and bytes will be counted as unicast.</p>
<code>show interface status</code>	Displays the interface line status.
<code>show interface vfi name</code>	Displays the status and information about the specified interface.
<code>show l2vpn atom vc</code>	Displays information about the Any Transport over MPLS (AToM) virtual circuit.
<code>show l2vpn service xconnect all</code>	Displays status information about the specified XConnect service.
<code>show mac address-table</code>	Displays the list of the known MAC addresses and their forwarding information

## Monitoring Tunnel Interfaces

You can configure DCNM to collect tunnel interface statistics. Choose **Interfaces > Logical > Tunnel** from the Feature Selector and navigate to the interface that you want to collect statistics on.

You see the Port Traffic Statistics window. You can collect statistics on input and output (packet and byte) counters, broadcast, multicast, and unicast traffic.

See the *Fundamentals Configuration Guide, Cisco DCNM for LAN, Release 5.x*, for more information on collecting statistics for layer 3 interfaces.

## Configuration Examples for Virtual Private LAN Service

This section includes the following topics:

- [Example: VPLS with a Bridge Domain, page 31-32](#)
- [Example: VPLS with a VLAN, page 31-32](#)
- [Example: VPLS Auto Discovery and BGP Signaling, page 31-33](#)
- [Example: VPLS Auto Discovery and LDP Signaling, page 31-33](#)
- [Example: VPLS with MPLS LDP, page 31-33](#)

## Example: VPLS with a Bridge Domain

The following example shows how to configure VPLS with a bridge domain configuration:

```
bridge-domain 100
  member vfi foo
  member Ethernet2/1 service instance 1
!
l2vpn vfi context foo
  vpn id 100
  member Pseudowire12
  member Pseudowire13
!
interface Pseudowire12 #mesh
  encapsulation mpls
  neighbor 10.2.2.2 100
!
interface Pseudowire13 #mesh
  encapsulation mpls
  neighbor 10.3.3.3 100
!
interface Ethernet2/1
  service instance 1 ethernet
  encapsulation dot1q 100
```

## Example: VPLS with a VLAN

The following example shows how to configure the same VPLS with a VLAN configuration:

```
vlan 100
vlan configuration 100
  member vfi foo
!
port-profile type pseudowire mpls
  encapsulation mpls
!
l2vpn vfi context foo
  vpn id 100
  member Pseudowire12
  member Pseudowire13
!
interface Pseudowire12 #mesh
  inherit port-profile mpls
  neighbor 10.2.2.2 100
!
interface Pseudowire13 #mesh
  inherit port-profile mpls
  neighbor 10.3.3.3 100
!
interface Ethernet2/1
  switchport mode trunk
```



```
switchport allowed vlan 100
```

## Example: VPLS Auto Discovery and BGP Signaling

The following example show how to configure VPLS auto discovery and BGP signaling:

```
Device bgp 100
  neighbor 10.0.0.2 remote-as 100
  address-family l2vpn vpls
    neighbor 10.0.0.2 activate
    neighbor 10.0.0.2 send-community extended
    neighbor 10.0.0.2 suppress-signaling-protocol ldp
  exit-address-family
```

## Example: VPLS Auto Discovery and LDP Signaling

The following example show how to configure VPLS auto discovery and LDP signaling:

```
bridge-domain 100
  member vfi foo
  member Ethernet2/1 service instance 1
!
l2vpn vfi context foo
  vpn id 100
  autodiscovery bgp signaling ldp
!
router bgp 100
  neighbor 10.0.0.1 remote-as 100
  address-family l2vpn vpls
!
interface Ethernet2/1
  service instance 1 ethernet
  encapsulation dot1q 100
```

## Example: VPLS with MPLS LDP

The following example show how to configure VPLS along with MPLS LDP between PE devices:

### PE1

```
feature-set mpls

feature ospf
feature mpls ldp
feature mpls l2vpn

l2vpn
feature evc

vlan 1,100

l2vpn vfi context foo
  vpn id 100
  member 20.0.0.4 encapsulation mpls

vlan configuration 100
  member vfi foo
```

```

interface Ethernet3/1
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 100
  no shutdown

interface Ethernet3/8
  mpls ip
ip address 11.1.1.1/24
  ip router ospf pe1 area 0.0.0.0
  no shutdown

interface loopback0
  ip address 20.0.0.1/32
  ip router ospf pe1 area 0.0.0.0
no terminal log-all
line vty
mpls ldp configuration
  discovery targeted-hello accept
  router-id Lo0 force
  neighbor 20.0.0.4 targeted
router ospf pe1

```

### Host P1

```

feature-set mpls
feature ospf
feature mpls ldp

interface Ethernet3/9
  mpls ip
  ip address 11.1.1.2/24
  ip router ospf p1 area 0.0.0.0
  no shutdown

interface Ethernet3/16
  mpls ip
  ip address 12.1.1.1/24
  ip router ospf p1 area 0.0.0.0
  no shutdown

interface loopback0
  ip address 20.0.0.2/32
  ip router ospf p1 area 0.0.0.0

mpls ldp configuration
  router-id Lo0 force
router ospf p1

```

### Host P2

```

feature-set mpls
feature ospf
feature mpls ldp

interface Ethernet3/17
  mpls ip
  ip address 12.1.1.2/24
  ip router ospf p2 area 0.0.0.0
  no shutdown

```

```
interface Ethernet3/32
  mpls ip
  ip address 13.1.1.1/24
  ip router ospf p2 area 0.0.0.0
  no shutdown

interface loopback0
  ip address 20.0.0.3/32
  ip router ospf p2 area 0.0.0.0

mpls ldp configuration
  router-id Lo0 force
router ospf p2
```

## PE2

```
feature-set mpls
feature ospf
feature mpls ldp
feature mpls l2vpn

l2vpn
feature evc

vlan 1,100

l2vpn vfi context foo
  vpn id 100
  member 20.0.0.1 encapsulation mpls

vlan configuration 100
  member vfi foo

interface Ethernet3/33
  mpls ip
  ip address 13.1.1.2/24
  ip router ospf pe2 area 0.0.0.0
  no shutdown

interface Ethernet3/47
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 100
  no shutdown

interface loopback0
  ip address 20.0.0.4/32
  ip router ospf pe2 area 0.0.0.0
no terminal log-all
line vty
mpls ldp configuration
  discovery targeted-hello accept
  router-id Lo0 force
  neighbor 20.0.0.3 targeted
router ospf pe2
```

# Additional References for Virtual Private LAN Service

For additional information related to configuring ACs for VPLS, see the following sections:

- [Related Documents](#), page 31-37
- [MIBs <Optional: remove if not applicable>](#), page 31-37

## Related Documents

Related Topic	Document Title
Interface commands	<i>Cisco Nexus 7000 Series NX-OS Interfaces Command Reference</i>
VLAN commands	<i>Cisco Nexus 7000 Series NX-OS Layer 2 Switching Command Reference</i>
Nondirectly connected MPLS LDP sessions	“Establishing Nondirectly Connected MPLS LDP Sessions” section of the “Configuring the MPLS Label Distribution Protocol” chapter.

## MIBs <Optional: remove if not applicable>

MIBs	MIBs Link
<ul style="list-style-type: none"> <li>• BRIDGE-MIB</li> <li>• CISCO-EVC-MIB</li> <li>• CISCO-VLAN-MEMBERSHIP-MIB</li> <li>• CISCO-IETF-PW-ATM-MIB (PW-ATM-MIB)</li> <li>• CISCO-IETF-PW-ENET-MIB (PW-ENET-MIB)</li> <li>• CISCO-IETF-PW-FR-MIB (PW-FR-MIB)</li> <li>• CISCO-IETF-PW-MIB (PW-MIB)</li> <li>• CISCO-IETF-PW-MPLS-MIB (PW-MPLS-MIB)</li> <li>• CISCO-VLAN-MEMBERSHIP-MIB</li> </ul>	<p>To locate and download MIBs, go to the following URL:</p> <p><a href="http://www.cisco.com/dc-os/mibs">http://www.cisco.com/dc-os/mibs</a></p>

## Feature History for Virtual Private LAN Service

Table 31-4 lists the release history for this feature.

**Table 31-4** Feature History for Virtual Private Lan Service

Feature Name	Releases	Feature Information
Virtual Private Lan Service (VPLS)	6.2(2)	This feature was introduced.  The following commands were introduced or modified: <b>address-family</b> , <b>autodiscovery bgp</b> , <b>bridge-domain</b> , <b>control-word</b> , <b>description</b> , <b>encapsulation</b> , <b>feature mpls l2vpn</b> , <b>interface pseudowire</b> , <b>l2vpn vfi context</b> , <b>member</b> , <b>member vfi</b> , <b>mtu</b> , <b>neighbor</b> , <b>router bgp</b> , <b>service instance</b> , <b>show interface</b> , <b>show interface pseudowire</b> , <b>show l2vpn atom vc</b> , <b>show l2vpn service vfi</b> , <b>show l2vpn vfi</b> , <b>switchport mode trunk</b> , <b>switchport allowed vlan</b> , <b>system bridge-domain</b> , <b>vlan</b> .
IP tunnels in VDC other than default	4.2(1)	This features was introduced.

