Virtual Private LAN Services (VPLS)

Note

- For complete syntax and usage information for the commands used in this chapter, see these publications:

  http://www.cisco.com/en/US/products/ps9536/prod_command_reference_list.html

- Cisco IOS Release 12.2SY supports only Ethernet interfaces. Cisco IOS Release 12.2SY does not support any WAN features or commands.

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Prerequisites for VPLS

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

- Configure IP routing in the core so that the PE routers can reach each other via IP.
- Configure MPLS in the core so that a label switched path (LSP) exists between the PE routers.
- Configure a loopback interface for originating and terminating Layer 2 traffic. Make sure the PE routers can access the other router's loopback interface. Note that the loopback interface is not needed in all cases. For example, tunnel selection does not need a loopback interface when VPLS is directly mapped to a TE tunnel.

VPLS configuration requires you to identify peer PE routers and to attach Layer 2 circuits to the VPLS at each PE router.

Restrictions for VPLS

- With a Supervisor Engine 2T, Layer 2 protocol tunneling is not supported with VPLS (CSCue45974).
- Split horizon is the default configuration to avoid broadcast packet looping and to isolate Layer 2 traffic. Split horizon prevents packets received from an emulated VC from being forwarded into another emulated VC. This technique is important for creating loop-free paths in a full-meshed network.
- Supported maximum values:
  - Total number of VFIs: 4,096 (4K)
  - Maximum combined number of edge and the core peer PEs per VFI:

    —VPLS: 250

    —H-VPLS 500
  - Total number of VC: 12,288 (12K)
- No software-based data plane is supported.
- No auto-discovery mechanism is supported.
- Load sharing and failover on redundant CE-PE links are not supported.
- The addition or removal of MAC addresses with Label Distribution Protocol (LDP) is not supported.
- The virtual forwarding instance (VFI) is supported only with the **interface vlan** command.

Information About VPLS

VPLS Overview

VPLS (Virtual Private LAN Service) enables enterprises to link together their Ethernet-based LANs from multiple sites via the infrastructure provided by their service provider. From the enterprise perspective, the service provider's public network looks like one giant Ethernet LAN. For the service provider, VPLS provides an opportunity to deploy another revenue-generating service on top of their existing network without major capital expenditures. Operators can extend the operational life of equipment in their network.

Virtual Private LAN Services (VPLS) uses the provider core to join multiple attachment circuits together to simulate a virtual bridge that connects the multiple attachment circuits together. From a customer point of view, there is no topology for VPLS. All of the CE devices appear to connect to a logical bridge emulated by the provider core (see Figure 37-1).

Figure 37-1 VPLS Topology

Full-Mesh Configuration

The full-mesh configuration requires a full mesh of tunnel label switched paths (LSPs) between all the PEs that participate in the VPLS. With full-mesh, signaling overhead and packet replication requirements for each provisioned VC on a PE can be high.

You set up a VPLS by first creating a virtual forwarding instance (VFI) on each participating PE router. The VFI specifies the VPN ID of a VPLS domain, the addresses of other PE routers in the domain, and the type of tunnel signaling and encapsulation mechanism for each peer PE router.

The set of VFIs formed by the interconnection of the emulated VCs is called a *VPLS instance*; it is the VPLS instance that forms the logic bridge over a packet switched network. The VPLS instance is assigned a unique VPN ID.

The PE routers use the VFI to establish a full-mesh LSP of emulated VCs to all the other PE routers in the VPLS instance. PE routers obtain the membership of a VPLS instance through static configuration using the Cisco IOS CLI.

The full-mesh configuration allows the PE router to maintain a single broadcast domain. Thus, when the PE router receives a broadcast, multicast, or unknown unicast packet on an attachment circuit, it sends the packet out on all other attachment circuits and emulated circuits to all other CE devices participating in that VPLS instance. The CE devices see the VPLS instance as an emulated LAN.

To avoid the problem of a packet looping in the provider core, the PE devices enforce a "split-horizon" principle for the emulated VCs. That means if a packet is received on an emulated VC, it is not forwarded on any other emulated VC.

After the VFI has been defined, it needs to be bound to an attachment circuit to the CE device.

The packet forwarding decision is made by looking up the Layer 2 virtual forwarding instance (VFI) of a particular VPLS domain.

A VPLS instance on a particular PE router receives Ethernet frames that enter on specific physical or logical ports and populates a MAC table similarly to how an Ethernet switch works. The PE router can use the MAC address to switch those frames into the appropriate LSP for delivery to the another PE router at a remote site.

If the MAC address is not in the MAC address table, the PE router replicates the Ethernet frame and floods it to all logical ports associated with that VPLS instance, except the ingress port where it just entered. The PE router updates the MAC table as it receives packets on specific ports and removes addresses not used for specific periods.

H-VPLS

Hierarchical VPLS (H-VPLS) reduces both signaling and replication overhead by using both full-mesh as well as hub and spoke configurations. Hub and spoke configurations operate with split horizon to allow packets to be switched between pseudo-wires (PWs), effectively reducing the number of PWs between PEs.

Note Split horizon is the default configuration to avoid broadcast packet looping. To avoid looping when using the **no-split-horizon** keyword, be very mindful of your network configuration.

Supported Features

- Multipoint-to-Multipoint Support, page 37-4
- Non-Transparent Operation, page 37-4
- Circuit Multiplexing, page 37-4
- MAC-Address Learning Forwarding and Aging, page 37-5
- Jumbo Frame Support, page 37-5
- Q-in-Q Support and Q-in-Q to EoMPLS Support, page 37-5
- VPLS Services, page 37-5

Multipoint-to-Multipoint Support

Two or more devices are associated over the core network. No one device is designated as the Root node, but all devices are treated as Root nodes. All frames can be exchanged directly between nodes.

Non-Transparent Operation

A virtual Ethernet connection (VEC) can be transparent or non-transparent with respect to Ethernet PDUs (that is, BPDUs). The purpose of VEC non-transparency is to allow the end user to have a Frame Relay-type service between Layer 3 devices.

Circuit Multiplexing

Circuit Multiplexing allows a node to participate in multiple services over a single Ethernet connection. By participating in multiple services, the Ethernet connection is attached to multiple logical networks. Some examples of possible service offerings are VPN services between sites, Internet services, and third-party connectivity for intercompany communications.

MAC-Address Learning Forwarding and Aging

PEs must learn remote MAC addresses and directly attached MAC addresses on customer facing ports. MAC address learning accomplishes this by deriving topology and forwarding information from packets originating at customer sites. A timer is associated with stored MAC addresses. After the timer expires, the entry is removed from the table.

Jumbo Frame Support

Jumbo frame support provides support for frame sizes between 1548 through 9216 bytes. You use the CLI to establish the jumbo frame size for any value specified in the above range. The default value is 1500 bytes in any Layer 2/VLAN interface. You can configure jumbo frame support on a per-interface basis.

Q-in-Q Support and Q-in-Q to EoMPLS Support

With 802.1Q tunneling (Q-in-Q), the CE issues VLAN-tagged packets and the VPLS forwards the packets to a far-end CE. Q-in-Q refers to the fact that one or more 802.1Q tags may be located in a packet within the interior of the network. As packets are received from a CE device, an additional VLAN tag is added to incoming Ethernet packets to segregate traffic from different CE devices. Untagged packets originating from the CE use a single tag within the interior of the VLAN switched network, while previously tagged packets originating from the CE use two or more tags.

VPLS Services

- Transparent LAN Service, page 37-5
- Ethernet Virtual Connection Service, page 37-6

Transparent LAN Service

Transparent LAN Service (TLS) is an extension to the point-to-point port-based EoMPLS, used to provide bridging protocol transparency (for example, bridge protocol data units [BPDUs]) and VLAN values. Bridges see this service as an Ethernet segment. With TLS, the PE router forwards all Ethernet packets received from the customer-facing interface (including tagged, untagged, and BPDUs) as follows:

- To a local Ethernet interface or an emulated VC if the destination MAC address is found in the Layer 2 forwarding table.
- To all other local Ethernet interfaces and emulated VCs belonging to the same VPLS domain if the destination MAC address is a multicast or broadcast address or if the destination MAC address is not found in the Layer 2 forwarding table.

Note With a Supervisor Engine 2T, Layer 2 protocol tunneling is not supported with VPLS, which prevents use of the Cisco Discovery Protocol (CDP), the VLAN Trunking Protocol (VTP), and the Spanning-Tree Protocol (STP) over VPLS (CSCue45974).

Ethernet Virtual Connection Service

Ethernet Virtual Connection Service (EVCS) is an extension to the point-to-point VLAN-based EoMPLS that allows routers to reach multiple intranet and extranet locations from a single physical port. Routers see subinterfaces through which they access other routers. With EVCS, the PE router forwards all Ethernet packets with a particular VLAN tag received from the customer-facing interface (excluding BPDUs) as follows:

- To a local Ethernet interface or to an emulated VC if the destination MAC address is found in the Layer 2 forwarding table.
- To all other local Ethernet interfaces and emulated VCs belonging to the same VPLS domain if the destination MAC address is a multicast or broadcast address or if the destination MAC address is not found in the Layer 2 forwarding table.

Note Because it has only local significance, the demultiplexing VLAN tag that identifies a VPLS domain is removed before forwarding the packet to the outgoing Ethernet interfaces or emulated VCs.

Default Settings for VPLS

None.

How to Configure VPLS

Note

- Use the procedures in the QoS chapters to configure QoS for VPLS traffic.
- Provisioning a VPLS link involves provisioning the associated attachment circuit and the VFI on the PE.

Configuring PE Layer 2 Interfaces to CEs

Note

- It is important to define the trunk VLANs; use the **switchport trunk allow vlan** command as shown in the first example below.
- You must configure the Layer 2 interface as a switchport for local bridging. You have the option of selecting tagged or untagged traffic from the CE device.

Configuring 802.1Q Trunks for Tagged Traffic from a CE

Note When EVCS is configured, the PE router forwards all Ethernet packets with a particular VLAN tag to a local Ethernet interface or emulated VC if the destination MAC address is found in the Layer 2 forwarding table.

	Command or Action	Purpose
Step 1	`Router(config)#` **`interface`** *`type number`*	Selects an interface to configure.
Step 2	`Router(config)#` **`no ip address`** *`ip_address mask`* `[`**`secondary`**`]`	Disables IP processing and enters interface configuration mode.
Step 3	`Router(config-if)#` **`switchport`**	Modifies the switching characteristics of the Layer 2-switched interface.
Step 4	`Router(config-if)#` **`switchport trunk encapsulation dot1q`**	Sets the switch port encapsulation format to 802.1Q.
Step 5	`Router(config-if)#` **`switchport trunk allow vlan`** *`vlan_ID`*	Sets the list of allowed VLANs.
Step 6	`Router(config-if)#` **`switchport mode trunk`**	Sets the interface to a trunking VLAN Layer 2 interface.

This example shows how to configure the tagged traffic.

```
Router(config)# interface GigabitEthernet4/4
Router(config)# no ip address
Router(config-if)# switchport
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# switchport trunk allow vlan 501
Router(config-if)# switchport mode trunk
```

This example shows how to use the **show run interface** command to verify the configuration.

```
Router# show run interface GigabitEthernet4/4
Building configuration...

Current configuration : 212 bytes
!
interface GigabitEthernet4/4
 no ip address
```

```
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 500-1999
 switchport mode trunk
end
```

Configuring 802.1Q Access Ports for Untagged Traffic from CE

	Command or Action	Purpose
Step 1	`Router(config)# interface type number`	Selects an interface to configure.
Step 2	`Router(config)# no ip address ip_address mask [secondary]`	Disables IP processing and enters interface configuration mode.
Step 3	`Router(config-if)# speed [1000 | nonegotiate]`	Sets the port speed for an Ethernet interface; enables or disables the link negotiation protocol on the Gigabit Ethernet ports.
Step 4	`Router(config-if)# switchport`	Modifies the switching characteristics of the Layer 2-switched interface.
Step 5	`Router(config-if)# switchport mode access`	Sets the interface type to nontrunking, nontagged single VLAN Layer 2 interface.
Step 6	`Router(config-if)# switchport access vlan vlan_id`	Sets the VLAN when the interface is in Access mode.

This example shows how to configure the untagged traffic.

```
Router(config)# interface GigabitEthernet4/4
Router(config)# no ip address
Router(config-if)# speed nonegotiate
Router(config-if)# switchport
Router(config-if)# switchport mode access
Router(config-if)# switchport access vlan 501
```

This example shows how to use the **show run interface** command to verify the configuration.

```
Router# show run interface GigabitEthernet4/4
Building configuration...

Current configuration : 212 bytes
!
interface GigabitEthernet4/4
 speed nonegotiate
 switchport
 switchport mode access
 switchport access vlan 501
end
```

Configuring Q-in-Q to Place All VLANs into a Single VPLS Instance

Note When TLS is configured, the PE router forwards all Ethernet packets received from the CE device to all local Ethernet interfaces and emulated VCs belonging to the same VPLS domain if the MAC address is not found in the Layer 2 forwarding table.

	Command or Action	Purpose
Step 1	`Router(config)#` **`interface`** *`type number`*	Selects an interface to configure.
Step 2	`Router(config)#` **`no ip address`** *`ip_address mask`* [**`secondary`**]	Disables IP processing and enters interface configuration mode.
Step 3	`Router(config-if)#` **`speed`** [**`1000`** | **`nonegotiate`**]	Sets the port speed for an Ethernet interface; enables or disables the link negotiation protocol on the Gigabit Ethernet ports.
Step 4	`Router(config-if)#` **`switchport`**	Modifies the switching characteristics of the Layer 2-switched interface.
Step 5	`Router(config-if)#` **`switchport access vlan`** *`vlan_id`*	Sets the VLAN when the interface is in Access mode.
Step 6	`Router(config-if)#` **`switchport mode dot1q-tunnel`**	Sets the interface as an 802.1Q tunnel port.
Step 7	`Router(config-if)#` **`l2protocol-tunnel`** [**`cdp`** | **`stp`** | **`vtp`**]	Enables protocol tunneling on an interface.

This example shows how to configure the tagged traffic.

```
Router(config)# interface GigabitEthernet4/4
Router(config)# no ip address
Router(config-if)# speed nonegotiate
Router(config-if)# switchport
Router(config-if)# switchport access VLAN 501
Router(config-if)# switchport mode dot1q-tunnel
Router(config-if)# l2protocol-tunnel cdp
```

This example shows how to use the **show run interface** command to verify the configuration.

```
Router# show run interface GigabitEthernet4/4
Building configuration...

Current configuration : 212 bytes
!
interface GigabitEthernet4/4
 no ip address
 speed nonegotiate
 switchport
 switchport access vlan 501
 switchport mode dot1q-tunnel
 l2protocol-tunnel cdp
end
```

Use the **show spanning-tree vlan** command to verify the port is not in a blocked state.

```
Router# show spanning-tree vlan 501

VLAN0501
Spanning tree enabled protocol ieee
  Root ID    Priority    33269
```

```
             Address     0001.6446.2300
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    33269  (priority 32768 sys-id-ext 501)
             Address     0001.6446.2300
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 0

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- --------
--------------------------------
Gi4/4            Desg FWD 4         128.388  P2p
```

Use the **show vlan id** command to verify that a specific port is configured to send and receive a specific VLAN's traffic.

```
Router# show vlan id 501

VLAN Name                             Status    Ports
---- -------------------------------- ---------
501  VLAN0501                         active    Gi4/4

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1
Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------
501  enet  100501     1500  -      -      -        -    -        0      0

Remote SPAN VLAN
----------------
Disabled

Primary Secondary Type              Ports
------- --------- -----------------
```

Configuring Layer 2 VLAN Instances on a PE

Configuring the Layer 2 VLAN interface on the PE enables the Layer 2 VLAN instance on the PE router to the VLAN database to set up the mapping between the VPLS and VLANs.

	Command or Action	Purpose
Step 1	**vlan** *vlan-id* Router(config)# vlan 809	Configures a specific virtual LAN (VLAN).
Step 2	**interface vlan** *vlan-id* Router(config)# interface vlan 501	Configures an interface on the VLAN.

This is an example of configuring a Layer 2 VLAN instance.

```
Router# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# vlan 501
Router(config)# interface vlan 501
Router(config-if)#
```

Use the **show interfaces vlan** command to verify the VLAN is in the up state (example not shown).

Configuring MPLS in the PE

To configure MPLS in the PE, you must provide the required MPLS parameters.

Note Before configuring MPLS, ensure that you have IP connectivity between all PEs by configuring Interior Gateway Protocol (IGP) (Open Shortes Path First [OSPF] or Intermediate System to Intermediate System [IS-IS]) between the PEs.

	Command or Action	Purpose
Step 1	**`enable`** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`mpls label protocol {ldp | tdp}`** `Router(config)# mpls label protocol ldp`	Specifies the default Label Distribution Protocol for a platform.
Step 4	**`mpls ldp logging neighbor-changes`** `Router(config)# mpls ldp logging neighbor-changes`	(Optional) Determines logging neighbor changes.
Step 5	`tag-switching tdp discovery {hello | directed hello} {holdtime | interval}` *`seconds`* `Router(config)# tag-switching tdp discovery hello holdtime 5`	Configures the interval between transmission of LDP (TDP) discovery hello messages, or the hold time for a LDP transport connection
Step 6	`tag-switching tdp router-id Loopback0 force` `Router(config)# tag-switching tdp router-id Loopback0 force`	Configures MPLS.

This example shows global MPLS configuration.

```
Router(config)# mpls label protocol ldp
Router(config)# tag-switching tdp discovery directed hello
Router(config)# tag-switching tdp router-id Loopback0 force
```

Use the **show ip cef** command to verify that the LDP label is assigned.

```
Router# show ip cef 192.168.17.7
192.168.17.7/32, version 272, epoch 0, cached adjacency to POS4/1
0 packets, 0 bytes
  tag information set
    local tag: 8149
    fast tag rewrite with PO4/1, point2point, tags imposed: {4017}
  via 11.3.1.4, POS4/1, 283 dependencies
    next hop 11.3.1.4, POS4/1
    valid cached adjacency
    tag rewrite with PO4/1, point2point, tags imposed: {4017}
```

Configuring the VFI in the PE

The virtual switch instance (VFI) specifies the VPN ID of a VPLS domain, the addresses of other PE routers in this domain, and the type of tunnel signaling and encapsulation mechanism for each peer. (This is where you create the VSI and associated VCs.) Configure a VFI as follows:

Note Only MPLS encapsulation is supported.

	Command or Action	Purpose

Step 1	`l2 vfi` *name* `manual`
	
	`Router(config)# l2 vfi vfi17 manual`
		Enables the Layer 2 VFI manual configuration mode.

Step 2	**`vpn id`** *vpn-id* `Router(config-vfi)# vpn id 17`	Configures a VPN ID for a VPLS domain. The emulated VCs bound to this Layer 2 VRF use this VPN ID for signaling.

Step 3	`neighbor` *remote router id* `{encapsulation mpls} [no-split-horizon]`
	
	`Router(config-vfi)# neighbor 1.5.1.1 encapsulation mpls`
		Specifies the remote peering router ID and the tunnel encapsulation type or the pseudo-wire property to be used to set up the emulated VC.
		
		Note Split horizon is the default configuration to avoid broadcast packet looping and to isolate Layer 2 traffic. Use the **no-split-horizo**n keyword to disable split horizon and to configure multiple VCs per spoke into the same VFI.

Step 4	`shutdown`
	
	`Router(config-vfi)# shutdown`
		Disconnects all emulated VCs previously established under the Layer 2 VFI and prevents the establishment of new attachment circuits.
		
		Note It does not prevent the establishment of new attachment circuits configured with the Layer 2 VFI using CLI.

The following example shows a VFI configuration.

```
Router(config)# l2 vfi VPLSA manual
Router(config-vfi)# vpn id 100
Router(config-vfi)# neighbor 11.11.11.11 encapsulation mpls
Router(config-vfi)# neighbor 33.33.33.33 encapsulation mpls
Router(config-vfi)# neighbor 44.44.44.44 encapsulation mpls
```

The following example shows a VFI configuration for hub and spoke.

```
Router(config)# l2 vfi VPLSA manual
Router(config-vfi)# vpn id 100
Router(config-vfi)# neighbor 9.9.9.9 encapsulation mpls
Router(config-vfi)# neighbor 12.12.12.12 encapsulation mpls
Router(config-vfi)# neighbor 33.33.33.33 encapsulation mpls no-split-horizon
```

The **show mpls 12transport vc** command displays various information related to PE1.

Note The **show mpls l2transport vc [detail]** command is also available to show detailed information about the VCs on a PE router as in the following example.

```
VPLS-PE2# show mpls l2transport vc 201

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
VFI test1      VFI                  153.1.0.1       201        UP
VFI test1      VFI                  153.3.0.1       201        UP
VFI test1      VFI                  153.4.0.1       201        UP
```

Note The VC ID in the output represents the VPN ID; the VC is identified by the combination of the Dest address and the VC ID as in the example below.

The **show vfi** *vfi name* command shows VFI status.

```
nPE-3# show vfi VPLS-2
VFI name: VPLS-2, state: up
  Local attachment circuits:
    Vlan2
  Neighbors connected via pseudowires:
  Peer Address     VC ID     Split-horizon
  1.1.1.1          2             Y
  1.1.1.2          2             Y
  2.2.2.3          2             N
```

Associating the Attachment Circuit with the VSI at the PE

After defining the VFI, you must bind it to one or more attachment circuits (interfaces, subinterfaces, or virtual circuits).

	Command or Action	Purpose
Step 1	**interface vlan** *vlan-id* Router(config-if)# interface vlan 100	Creates or accesses a dynamic switched virtual interface (SVI).
Step 2	no ip address Router(config-if)# no ip address	Disables IP processing. (You configure a Layer 3 interface for the VLAN if you configure an IP address.)
Step 3	**xconnect vfi** *vfi name* Router(config-if)# xconnect vfi vfi16	Specifies the Layer 2 VFI that you are binding to the VLAN port.

This example shows an interface VLAN configuration.

```
Router(config-if)# interface vlan 100
Router(config-if)# no ip address
Router(config-if)# xconnect vfi VPLS_501
```

Use the **show vfi** command for VFI status.

```
Router# show vfi VPLS_501
VFI name: VPLS_501, state: up
  Local attachment circuits:
    vlan 100
  Neighbors connected via pseudowires:
    192.168.11.1  192.168.12.2  192.168.13.3  192.168.16.6
    192.168.17.7
```

H-VPLS with MPLS Edge

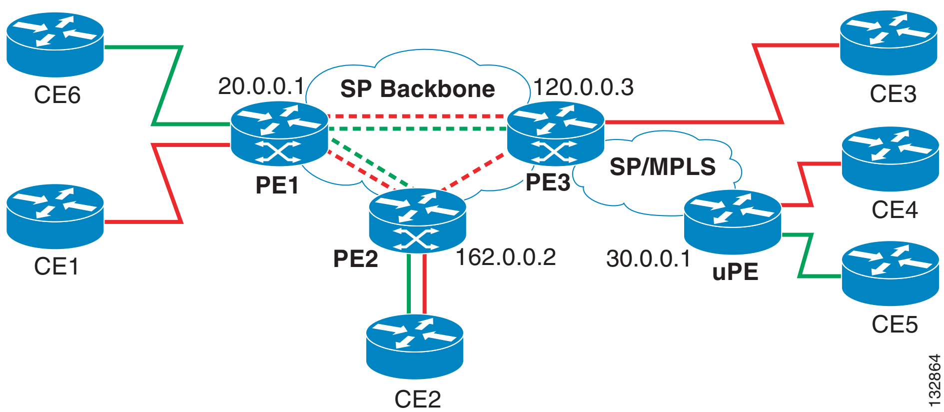
Overview

The Hierarchical VPLS model comprises hub and spoke and full-mesh networks. In a full-mesh configuration, each PE router creates a multipoint-to-multipoint forwarding relationship with all other PE routers in the VPLS domain using VFIs.

In the hub and spoke configuration, a PE router can operate in a non-split-horizon mode that allows inter-VC connectivity without the requirement to add a Layer 2 port in the VLAN.

In the example below, the VLANs on CE1, CE2, CE3, and CE4 (in red) connect through a full-mesh network. The VLANs on CE2, CE5, and ISP POP connect through a hub and spoke network where the ISP POP is the hub and CE2 and CE5 are the spokes. Figure 37-2 shows the configuration example.

Figure 37-2 H-VPLS Configuration



Configuration on PE1

Configuring VSIs and VCs

This sample configuration shows the creation of the virtual switch instances (VSIs) and associated VCs. Note that the VCs in green require the **no-split-horizon** keyword. The **no-split-horizon** command disables the default Layer 2 split horizon in the data path.

```
l2 vfi Internet manual
 vpn id 100
 neighbor 120.0.0.3 encapsulation mpls no-split-horizon
 neighbor 162.0.0.2 encapsulation mpls no-split-horizon

l2 vfi PE1-VPLS-A manual
 vpn id 200
 neighbor 120.0.0.3 encapsulation mpls
 neighbor 162.0.0.2 encapsulation mpls

interface Loopback 0
 ip address 20.0.0.1 255.255.255.255
```

Configuring the CE Device Interface

This sample configuration shows the CE device interface (there can be multiple Layer 2 interfaces in a VLAN).

```
interface GigEthernet1/1
 switchport
 switchport mode trunk
 switchport trunk encap dot1q
 switchport trunk allow vlan 1001,1002-1005
```

Associating the Attachment Circuit with the VFI

This sample configuration shows how the attachment circuit (VLAN) is associated with the VFI.

```
interface Vlan 1001
 xconnect vfi Internet

interface FastEthernet2/1
 switchport
 switchport mode trunk
 switchport trunk encap dot1q
 switchport trunk allow vlan 211,1002-1005

interface Vlan 211
 xconnect vfi PE1-VPLS-A
```

Configuration on PE2

- Configuring VSIs and VCs, page 37-15
- Configuring the CE Device Interface, page 37-16
- Associating the Attachment Circuit with the VFI, page 37-16

Configuring VSIs and VCs

This sample configuration shows the creation of the virtual switch instances (VSIs) and associated VCs.

```
l2 vfi Internet manual
 vpn id 100
 neighbor 20.0.0.1 encapsulation mpls
```

```
l2 vfi PE2-VPLS-A manual
 vpn id 200:1
 neighbor 120.0.0.3 encapsulation mpls
 neighbor 20.0.0.1 encapsulation mpls

interface Loopback 0
 ip address 162.0.0.2 255.255.255.255
```

Configuring the CE Device Interface

This sample configuration shows the CE device interface (there can be multiple Layer 2 interfaces in a VLAN).

```
interface GigEthernet2/1
 switchport
 switchport mode trunk
 switchport trunk encap dot1q
 switchport trunk allow vlan 211,1001,1002-1005
```

Associating the Attachment Circuit with the VFI

This sample configuration shows how the attachment circuit (VLAN) is associated with the VFI.

```
interface Vlan 1001
 xconnect vfi Internet

interface Vlan 211
 xconnect vfi PE2-VPLS-A
```

Configuration on PE3

- Configuring VSIs and VCs, page 37-16
- Configuring the CE Device Interface, page 37-17
- Configuring the Attachment Circuits, page 37-17
- Configuring Port-based EoMPLS on the uPE Device, page 37-17

Configuring VSIs and VCs

This sample configuration shows the creation of the virtual switch instances (VSIs) and associated VCs.

```
l2 vfi Internet manual
 vpn id 100
 neighbor 20.0.0.1 encapsulation mpls
 neighbor 162.0.0.2 encapsulation mpls
 neighbor 30.0.0.1 encapsulation mpls no-split horizon

l2 vfi PE3-VPLS-A manual
 vpn id 200
 neighbor 162.0.0.2 encapsulation mpls
 neighbor 20.0.0.1 encapsulation mpls

interface Loopback 0
 ip address 120.0.0.3 255.255.255.255
```

Configuring the CE Device Interface

This sample configuration shows the CE device interface (there can be multiple Layer 2 interfaces in a VLAN).

```
interface GigEthernet6/1
 switchport
 switchport mode trunk
 switchport trunk encap dot1q
 switchport trunk allow vlan 211
```

Configuring the Attachment Circuits

This sample configuration shows the attachment circuits.

```
interface Vlan 1001
 xconnect vfi Internet

interface Vlan 211
 xconnect vfi PE3-VPLS-A
```

Configuring Port-based EoMPLS on the uPE Device

This sample configuration shows port-based EoMPLS on the uPE device.

```
interface GigEthernet 1/1
 xconnect 120.0.0.3 100 encapsulation mpls
```

VPLS Integrated Routing and Bridging

VPLS integrated routing and bridging can route Layer 3 traffic as well as switch Layer 2 frames for pseudowire connections between provider edge (PE) devices using Virtual Private LAN Services (VPLS) multipoint PE. The ability to route frames to and from these interfaces supports termination of a pseudowire into a Layer 3 network (VPN or global) on the same switch, or to tunnel Layer 3 frames over a Layer 2 tunnel (VPLS).

Note

- VPLS integrated routing and bridging is also known as routed pseudowire and routed VPLS.
- VPLS integrated routing and bridging does not support multicast routing.

To configure routing support for the pseudowire, configure an IP address and other Layer 3 features for the Layer 3 domain (VPN or global) in the virtual LAN (VLAN) interface configuration.

- The following example assigns the IP address 10.10.10.1 to the VLAN 100 interface. (Layer 2 forwarding is defined by the VFI VFI100.)

```
interface vlan 100
  xconnect vfi VFI100
  ip address 10.10.10.1 255.255.255.0
```

- The following example assigns an IP address 20.20.20.1 of the VPN domain VFI200. (Layer 2 forwarding is defined by the VFI VFI200.)

```
interface vlan 200
  xconnect vfi VFI200
  ip vrf forwarding VFI200
  ip address 20.20.20.1 255.255.255.0
```

Configuration Examples for VPLS

In a full-mesh configuration, each PE router creates a multipoint-to-multipoint forwarding relationship with all other PE routers in the VPLS domain using a VFI. An Ethernet or VLAN packet received from the customer network can be forwarded to one or more local interfaces and or emulated VCs in the VPLS domain. To avoid broadcasted packets looping around in the network, no packet received from an emulated VC can be forwarded to any emulated VC of the VPLS domain on a PE router. That is, the Layer 2 split horizon should always be enabled as the default in a full-mesh network.

Figure 37-3 VPLS Configuration Example

Configuration on PE 1

This shows the creation of the virtual switch instances (VSIs) and associated VCs.

```
l2 vfi PE1-VPLS-A manual
  vpn id 100
  neighbor 2.2.2.2 encapsulation mpls
  neighbor 3.3.3.3 encapsulation mpls
!
interface Loopback 0
  ip address 1.1.1.1 255.255.255.255
```

This configures the CE device interface (there can be multiple Layer 2 interfaces in a VLAN).

```
interface FastEthernet0/0
  switchport
  switchport mode dot1qtunnel
  switchport access vlan 100
```

Here the attachment circuit (VLAN) is associated with the VSI.

```
interface vlan 100
  no ip address
 xconnect vfi PE1-VPLS-A
```

This is the enablement of the Layer 2 VLAN instance.

```
vlan 100
  state active
```

Configuration on PE 2

This shows the creation of the virtual switch instances (VSIs) and associated VCs.

```
l2 vfi PE2-VPLS-A manual
```

```
  vpn id 100
  neighbor 1.1.1.1 encapsulation mpls
  neighbor 3.3.3.3 encapsulation mpls
!
interface Loopback 0
  ip address 2.2.2.2 255.255.255.255
```

This configures the CE device interface (there can be multiple Layer 2 interfaces in a VLAN).

```
interface FastEthernet0/0
  switchport
  switchport mode dot1qtunnel
  switchport access vlan 100
```

Here the attachment circuit (VLAN) is associated with the VSI.

```
interface vlan 100
  no ip address
 xconnect vfi PE2-VPLS-A
```

This is the enablement of the Layer 2 VLAN instance.

```
vlan 100
  state active
```

Configuration on PE 3

This shows the creation of the virtual switch instances (VSIs) and associated VCs.

```
l2 vfi PE3-VPLS-A manual
  vpn id 100
  neighbor 1.1.1.1 encapsulation mpls
  neighbor 2.2.2.2 encapsulation mpls
!
interface Loopback 0
  ip address 3.3.3.3 255.255.255.255
```

This configures the CE device interface (there can be multiple Layer 2 interfaces in a VLAN).

```
interface FastEthernet0/1
  switchport
  switchport mode dot1qtunnel
  switchport access vlan 100
!
```

Here the attachment circuit (VLAN) is associated with the VSI.

```
interface vlan 100
  no ip address
 xconnect vfi PE3-VPLS-A .
!
```

This is the enablement of the Layer 2 VLAN instance.

```
vlan 100
  state active
```

The **show mpls l2 vc** command provides information on the status of the VC.

```
VPLS1# show mpls l2 vc

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
Vi1            VFI                  22.22.22.22     100        DOWN
Vi1            VFI                  22.22.22.22     200        UP
Vi1            VFI                  33.33.33.33     100        UP
Vi1            VFI                  44.44.44.44     100        UP
Vi1            VFI                  44.44.44.44     200        UP
```

The **show vfi** command provides information on the VFI.

```
PE-1# show vfi PE1-VPLS-A
VFI name: VPLSA, state: up
  Local attachment circuits:
    Vlan100
  Neighbors connected via pseudowires:
    2.2.2.2  3.3.3.3
```

The **show mpls 12transport vc** command provides information the virtual circuits.

```
Router# show mpls l2 vc det
Local interface: VFI vfi17 up
  Destination address: 1.3.1.1, VC ID: 17, VC status: up
    Tunnel label: imp-null, next hop point2point
    Output interface: PO3/4, imposed label stack {18}
  Create time: 3d15h, last status change time: 1d03h
  Signaling protocol: LDP, peer 1.3.1.1:0 up
    MPLS VC labels: local 18, remote 18
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 0, send 0
    byte totals:   receive 0, send 0
    packet drops:  receive 0, send 0
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum