H-VPLS N-PE Redundancy for QinQ Access

The H-VPLS N-PE Redundancy for QinQ Access feature enables two network provider edge (N-PE) devices to provide failover services to a user provider edge (U-PE) device in a hierarchical virtual private LAN service (H-VPLS). Having redundant N-PE devices provides improved stability and reliability against link and node failures.

Prerequisites for H-VPLS N-PE Redundancy for QinQ Access

- Before configuring this feature, configure your hierarchical virtual private LAN service (H-VPLS) network and make sure it is operating correctly.
- Make sure that the PE-to-customer edge (CE) interface is configured with a list of allowed VLANs.
- To provide faster convergence, you can enable the MPLS Traffic Engineering—Fast Reroute feature in the Multiprotocol Label Switching (MPLS) core.
- Enable the L2VPN Pseudowire Redundancy feature on the user provider edge (U-PE) devices for MPLS access.
- When configuring Multiple Spanning Tree Protocol (MSTP), specify that one of the network provider edge (N-PE) devices is the root by assigning it the lowest priority using the **spanning-tree mst** *instance-id* **priority** *priority* command.
- When configuring MSTP, make sure that each device participating in the spanning tree is in the same region and is the same revision by issuing the **revision**, **name**, and **instance** commands in MST configuration mode.

Restrictions for H-VPLS N-PE Redundancy for QinQ Access

- This feature cannot be used with the VPLS Autodiscovery feature on pseudowires that attach to network provider edge (N-PE) devices. When you create the virtual private LAN service (VPLS), you can manually create the virtual forwarding instance (VFI).
- You cannot configure more than one pseudowire to carry the bridge protocol data unit (BPDU) packets between two redundant network provider edge (N-PE) devices on the same Virtual Private LAN service (VPLS) site.
- You cannot configure a local loopback address as a neighbor when you configure the H-VPLS N-PE Redundancy feature on N-PE devices. If you do so, the following error message is displayed:

```
VPLS local switching to peer address not supported
```

- Only two N-PE devices can be connected to each U-PE device.
- The spanning-tree mode must be Multiple Spanning Tree Protocol (MSTP) for the H-VPLS N-PE Redundancy feature. If the spanning-tree mode changes, the H-VPLS N-PE Redundancy feature might not work correctly, even though the pseudowire that carries the BPDU packet still exists and the H-VPLS N-PE Redundancy feature is still configured.

Information About H-VPLS N-PE Redundancy for QinQ Access

How H-VPLS N-PE Redundancy for QinQ Access Works

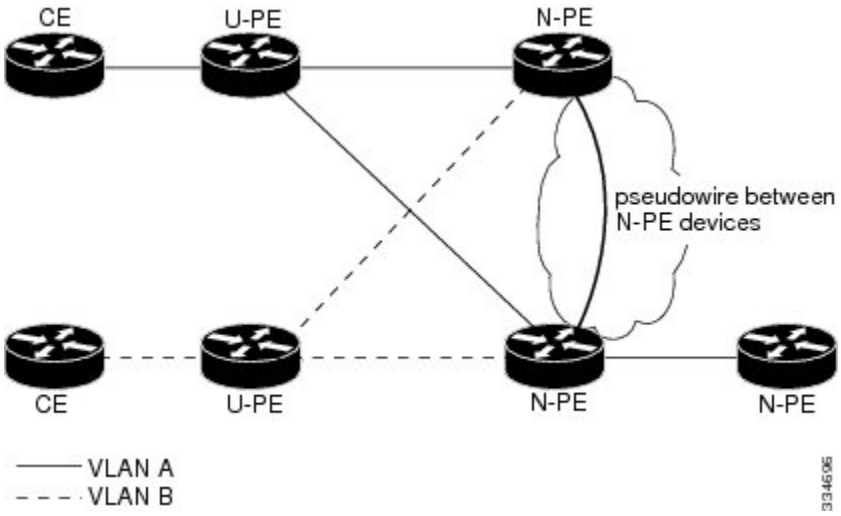
In a network configured with the H-VPLS N-PE Redundancy feature, the user provider edge (U-PE) device is connected to two network provider edge (N-PE) devices. This feature provides a level of redundancy that can tolerate both link and device faults. If a failure occurs in the network that disables one N-PE device from transmitting data, the other N-PE device takes over. This feature works with both QinQ access based on Multiple Spanning Tree Protocol (MSTP) and Multiprotocol Label Switching (MPLS) access based on pseudowire redundancy.

H-VPLS N-PE Redundancy with QinQ Access Based on MSTP

The H-VPLS N-PE Redundancy with QinQ Access feature uses the Multiple Spanning Tree Protocol (MSTP) running on the network provider edge (N-PE) devices and user provider edge (U-PE) devices in a hierarchical Virtual Private LAN service (H-VPLS) network. A pseudowire running between N-PE devices carries only MSTP bridge protocol data units (BPDUs). The pseudowire running between the N-PE devices is always up and is used to create a loop path between N-PE devices so that MSTP blocks one of the redundant paths between the U-PE device and the N-PE devices. If the primary N-PE device or the path to it fails, MSTP enables the path to the backup N-PE device.

The figure below shows an H-VPLS network with redundant access. Each U-PE device has two connections, one to each N-PE device. Between the two N-PE devices is a pseudowire to provide a loop path for MSTP BPDUs. The network topology allows for the backup N-PE device to take over if the primary N-PE device or the path to it fails.

Figure 1: H-VPLS N-PE Redundancy with QinQ Access Based on MSTP



How to Configure H-VPLS N-PE Redundancy for QinQ Access

Configuring the VPLS Pseudowire Between the N-PE Devices using the commands associated with the L2VPN Protocol-Based CLIs feature

Configuring network provider edge (N-PE) redundancy in a hierarchical Virtual Private LAN service (H-VPLS) network requires that you configure the VPLS pseudowire for transporting bridge protocol data unit (BPDU) packets. For the core pseudowire between the N-PE devices, you configure a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) and attach the VFI to a bridge-domain (described here). Then, in the next task, you bind the service instance to the bridge-domain. This configuration provides a redundancy that provides improved reliability against link and node failures.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *name*
4. **vpn id** *vpn id*
5. **member** *ip-address* **encapsulation mpls**
6. **forward permit l2protocol all**
7. **exit**
8. **bridge-domain** *bridge-id*
9. **member vfi** *vfi-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn vfi context** *name* **Example:** `Device(config)# l2vpn vfi context VPLS-10`	Establishes a L2VPN VFI between two or more separate networks, and enters L2VFI configuration mode.
Step 4	**vpn id** *vpn id* **Example:** `Device(config-vfi)# vpn id 10`	Sets a VPN ID on the Virtual Private LAN Services (VPLS) instance. • Use the same VPN ID for the PE devices that belong to the same VPN. • Make sure the VPN ID is unique for each VPN in the service provider network. The range is from 1 to 4294967295.
Step 5	**member** *ip-address* **encapsulation mpls** **Example:** `Device(config-vfi)# member 102.102.102.102 encapsulation mpls`	Specifies the devices that form a point-to-point L2VPN VFI connection. • *ip-address*—IP address of the VFI neighbor. • **encapsulation mpls**—Specifies Multiprotocol Label Switching (MPLS) as the data encapsulation method.
Step 6	**forward permit l2protocol all** **Example:** `Device(config-vfi)# forward permit l2protocol all`	Creates a pseudowire that is to be used to transport BPDU packets between the two N-PE devices.
Step 7	**exit** **Example:** `Device(config-vfi)# exit`	Returns to global configuration mode.
Step 8	**bridge-domain** *bridge-id* **Example:** `Device(config)# bridge-domain 10`	Configures components on a bridge domain, and enters bridge-domain configuration mode.
Step 9	**member vfi** *vfi-name* **Example:** `Device(config-bdomain)# member vfi VPLS-10`	Configures the VFI member in the bridge-domain.

	Command or Action	Purpose
Step 10	**end** **Example:** `Device(config-bdomain)# end`	Returns to privileged EXEC mode.

Configuring the VPLS Pseudowire Between the N-PE Devices using the commands associated with the L2VPN Protocol-Based CLIs feature

Configuring network provider edge (N-PE) redundancy in a hierarchical Virtual Private LAN service (H-VPLS) network requires that you configure the VPLS pseudowire for transporting bridge protocol data unit (BPDU) packets. For the core pseudowire between the N-PE devices, you configure a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) and attach the VFI to a bridge-domain (described here). Then, in the next task, you bind the service instance to the bridge-domain. This configuration provides a redundancy that provides improved reliability against link and node failures.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *name*
4. **vpn id** *vpn id*
5. **member** *ip-address* **encapsulation mpls**
6. **forward permit l2protocol all**
7. **exit**
8. **bridge-domain** *bridge-id*
9. **member vfi** *vfi-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn vfi context** *name* **Example:** `Device(config)# l2vpn vfi context VPLS-10`	Establishes a L2VPN VFI between two or more separate networks, and enters L2VFI configuration mode.

	Command or Action	Purpose
Step 4	**vpn id** *vpn id* **Example:** `Device(config-vfi)# vpn id 10`	Sets a VPN ID on the Virtual Private LAN Services (VPLS) instance. • Use the same VPN ID for the PE devices that belong to the same VPN. • Make sure the VPN ID is unique for each VPN in the service provider network. The range is from 1 to 4294967295.
Step 5	**member** *ip-address* **encapsulation mpls** **Example:** `Device(config-vfi)# member 102.102.102.102 encapsulation mpls`	Specifies the devices that form a point-to-point L2VPN VFI connection. • *ip-address*—IP address of the VFI neighbor. • **encapsulation mpls**—Specifies Multiprotocol Label Switching (MPLS) as the data encapsulation method.
Step 6	**forward permit l2protocol all** **Example:** `Device(config-vfi)# forward permit l2protocol all`	Creates a pseudowire that is to be used to transport BPDU packets between the two N-PE devices.
Step 7	**exit** **Example:** `Device(config-vfi)# exit`	Returns to global configuration mode.
Step 8	**bridge-domain** *bridge-id* **Example:** `Device(config)# bridge-domain 10`	Configures components on a bridge domain, and enters bridge-domain configuration mode.
Step 9	**member vfi** *vfi-name* **Example:** `Device(config-bdomain)# member vfi VPLS-10`	Configures the VFI member in the bridge-domain.
Step 10	**end** **Example:** `Device(config-bdomain)# end`	Returns to privileged EXEC mode.

Binding the Service Instance to the Bridge-Domain

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance** *id* **ethernet**

5. **encapsulation dot1q** *vlan-id*
6. **exit**
7. **bridge-domain** *bridge-id*
8. **member** *interface-type-number* **service-instance** *service-id*
9. **end**

DETAILED STEPS

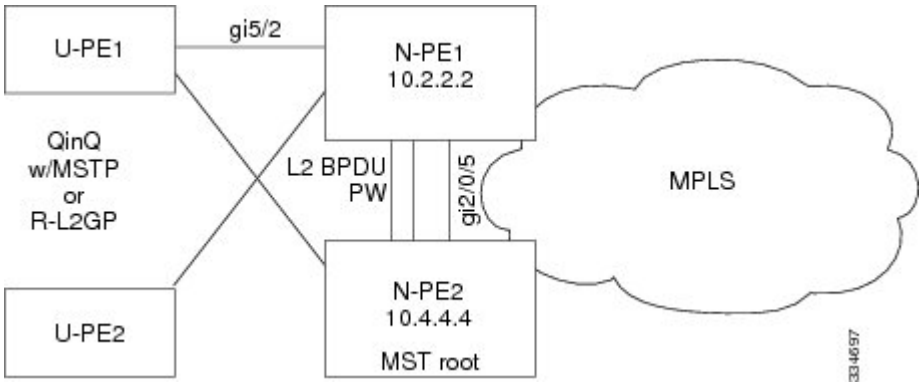
	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/1/0`	Specifies the interface to configure, and enters interface configuration mode.
Step 4	**service instance** *id* **ethernet** **Example:** `Device(config-if)# service instance 10 ethernet`	Configures an Ethernet service instance on the interface, and enters Ethernet service configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-if-srv)# encapsulation dot1q 10`	Enables IEEE 802.1Q encapsulation of traffic on the specified interface in a VLAN.
Step 6	**exit** **Example:** `Device(config-if-srv)# exit`	Returns to global configuration mode.
Step 7	**bridge-domain** *bridge-id* **Example:** `Device(config)# bridge-domain 10`	Configures components on the bridge domain, and enters bridge-domain configuration mode.
Step 8	**member** *interface-type-number* **service-instance** *service-id* **Example:** `Device(config-bdomain)# member GigabitEthernet0/1/0 service-instance 10`	Binds the service instance to the bridge-domain instance.
Step 9	**end** **Example:** `Device(config-bdomain)# end`	Returns to privileged EXEC mode.

Configuration Examples for H-VPLS N-PE Redundancy for QinQ Access

Example: H-VPLS N-PE Redundancy for QinQ Access

The figure below shows a configuration that is set up for the H-VPLS N-PE Redundancy with QinQ Access feature.

Figure 2: H-VPLS N-PE Redundancy with QinQ Access Topology



The table below shows the configuration of two network provider edge (N-PE) devices.

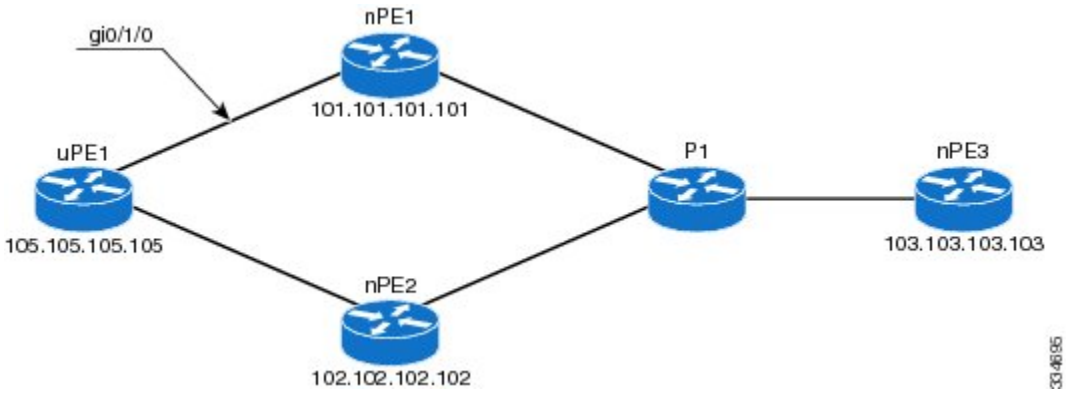
Table 1: Example: H-VPLS N-PE Redundancy for QinQ Access

N-PE1	N-PE2
l2vpn vfi context VPLS-10 vpn id 10 member 10.4.4.4 encapsulation mpls forward permit l2protocol all ! bridge-domain 10 member vfi VPLS-10 member GigabitEthernet5/2 service-instance 10 ! interface GigabitEthernet5/2 service instance 10 ethernet encapsulation dot1q 10 ! spanning-tree mode mst spanning-tree extend system-id ! spanning-tree mst configuration name myMstName revision 10 instance 1 vlan 10	l2vpn vfi context VPLS-10 vpn id 10 member 10.2.2.2 encapsulation mpls forward permit l2protocol all ! bridge-domain 10 member vfi VPLS-10 member GigabitEthernet2/0/5 service-instance 10 ! interface GigabitEthernet2/0/5 service instance 10 ethernet encapsulation dot1q 10 ! spanning-tree mode mst spanning-tree extend system-id ! spanning-tree mst configuration name myMstName revision 10 instance 1 vlan 20 ! spanning-tree mst 1 priority 0

Example: H-VPLS N-PE Redundancy for MPLS Access using the commands associated with the L2VPN Protocol-Based CLIs feature

The figure below shows a configuration that is set up for the H-VPLS N-PE Redundancy with MPLS Access feature. Because there is no option to configure multihoming on access VPLS, the **xconnect** command is used with priority on uPE1.

Figure 3: H-VPLS N-PE Redundancy with MPLS Access Topology



nPE1 Configuration

```
l2vpn vfi context VPLS-10
 vpn id 10
 member 102.102.102.102 encapsulation mpls
 member 103.103.103.103 encapsulation mpls
!
bridge-domain 10
 member vfi VPLS-10
 member 105.105.105.105 10 encapsulation mpls
```

nPE2 Configuration

```
l2vpn vfi context VPLS-10
 vpn id 10
 member 101.101.101.101 encapsulation mpls
 member 103.103.103.103 encapsulation mpls
!
bridge-domain 10
 member vfi VPLS-10
 member 105.105.105.105 10 encapsulation mpls
```

nPE3 Configuration

```
l2vpn vfi context VPLS-10
 vpn id 10
 member 101.101.101.101 encapsulation mpls
 member 102.102.102.102 encapsulation mpls
!
bridge-domain 10
 member vfi VPLS-10
```

uPE1 Configuration

```
interface GigabitEthernet0/1/0
 service instance 10 ethernet
 encapsulation dot1q 10
!
l2vpn xconnect context XC-10
 member GigabitEthernet0/1/0 service-instance 10
 member 101.101.101.101 10 encapsulation mpls group pwred priority 9
 member 102.102.102.102 10 encapsulation mpls group pwred priority 10
```

Sample Output on uPE1

```
Device# show l2vpn service peer 101.101.101.101 vcid 10

Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority
        UP=Up       DN=Down             AD=Admin Down    IA=Inactive
        SB=Standby  HS=Hot Standby      RV=Recovering    NH=No Hardware
        m=manually selected

  Interface          Group       Encapsulation                  Prio  St  XC St
  ---------          -----       -------------                  ----  --  -----
VPWS name: foo, State: UP
  Eth1/1.1                       Eth1/1.1:100(Eth VLAN)         0     UP  UP
  pw101              blue        102.1.1.1:100(MPLS)            2     UP  UP
  pw102              blue        103.1.1.1:100(MPLS)            5     SB  IA
  pw103              blue        104.1.1.1:100(MPLS)            8     SB  IA
  pw104              blue        105.1.1.1:100(MPLS)            11    SB  IA

Device# show l2vpn service peer 102.102.102.102 vcid 10

Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority
        UP=Up       DN=Down             AD=Admin Down    IA=Inactive
        SB=Standby  HS=Hot Standby      RV=Recovering    NH=No Hardware
        m=manually selected

  Interface          Group       Encapsulation                  Prio  St  XC St
  ---------          -----       -------------                  ----  --  -----
VPWS name: foo, State: UP
  Eth1/1.1                       Eth1/1.1:100(Eth VLAN)         0     UP  UP
  pw101              blue        102.1.1.1:100(MPLS)            2     UP  UP
  pw102              blue        103.1.1.1:100(MPLS)            5     SB  IA
  pw103              blue        104.1.1.1:100(MPLS)            8     SB  IA
  pw104              blue        105.1.1.1:100(MPLS)            11    SB  IA
```

Additional References for L2VPN VPLS Inter-AS Option B

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference

Related Topic	Document Title
IP Routing (BGP) commands	Cisco IOS IP Routing: BGP Command Reference
Concepts and tasks related to configuring the VPLS Autodiscovery: BGP Based feature.	*VPLS Autodiscovery BGP Based*
BGP support for the L2VPN address family	*BGP Support for the L2VPN Address Family*
VPLS	"VPLS Overview" section in the *Configuring Multiprotocol Label Switching on the Optical Services Modules* document
L2VPN multisegment pseudowires, MPLS OAM support for L2VPN multisegment pseudowires, MPLS OAM support for L2VPN inter-AS option B	*L2VPN Multisegment Pseudowires*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing standards has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 4360	*BGP Extended Communities Attribute*
RFC 4364	*BGP/MPLS IP Virtual Private Networks (VPNs)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for H-VPLS N-PE Redundancy for QinQ Access

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2: Feature Information for H-VPLS N-PE Redundancy for QinQ Access

Feature Name	Releases	Feature Information
H-VPLS N-PE Redundancy for QinQ Access	12.2(33)SRC 12.2(50)SY Cisco IOS XE Release 3.8S	The H-VPLS N-PE Redundancy for QinQ Access feature provides the capability to dual-home a given user provider edge (U-PE) device to two network provide edge (N-PE) devices in order to provide protection against link and node failures. In Cisco IOS Release 12.2(33)SRC, this feature was introduced on the Cisco 7600 series routers. In Cisco IOS Release 12.2(50)SY, this feature was integrated. In Cisco IOS XE Release 3.8S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **forward permit l2protocol**, **show mpls l2transport vc**.

Glossary

CE device—customer edge device. A device that belongs to a customer network, which connects to a PE device to utilize MPLS VPN network services.

LAN—local-area network. High-speed, low-error data network covering a relatively small geographic area. LANs connect workstations, peripherals, terminals, and other devices in a single building or other geographically limited areas.

MPLS—Multiprotocol Label Switching. A packet-forwarding technology, used in the network core, that applies data link layer labels to tell switching nodes how to forward data, resulting in faster and more scalable forwarding than network layer routing normally can do.

MSTP—Multiple Spanning Tree Protocol. MSTP enables multiple VLANs to be mapped to the same spanning-tree instance, reducing the number of spanning-tree instances needed to support a large number of VLANs.

N-PE—network provider edge device. This device acts as a gateway between the MPLS core and edge domains.

PE device—provider edge device. The PE device is the entry point into the service provider network. The PE device is typically deployed on the edge of the network and is administered by the service provider.

pseudowire—A pseudowire is a virtual connection that, in the context of VPLS, connects two SVIs. It is a mechanism that carries the elements of an emulated service from one PE device to one or more PE devices over a packet switched network (PSN). A pseudowire is bidirectional and consists of a pair of unidirectional MPLS virtual circuits (VCs). A pseudowire can be used to connect a point-to-point circuit.

QinQ—An IEEE 802.1Q VLAN tunnel. A mechanism for constructing multipoint Layer 2 VPN using Ethernet switches.

redundancy—The duplication of devices, services, or connections so that, in the event of a failure, they can perform the work of those that failed.

router—A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

spanning tree—Loop-free subset of a network topology.

U-PE—user provider edge device. This device connects CE devices to the service.

VFI—virtual forwarding instance. A VFI is a collection of data structures used by the data plane, software-based or hardware-based, to forward packets to one or more VCs.

VLAN—Virtual LAN. Group of devices on one or more LANs that are configured (using management software) so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments.

VPLS—Virtual Private LAN Service. VPLS describes an architecture that delivers Layer 2 service that emulates an Ethernet LAN across a wide-area network (WAN) and inherits the scaling characteristics of a LAN.

VPLS redundancy—Also called N-PE redundancy. Allows U-PEs to be dual-honed (to their N-PEs) in a loop-free topology with MPLS or QinQ as the access or aggregation domain.

VPN—Virtual Private Network. Allows IP traffic to travel securely over public TCP/IP networks and the Internet by encapsulating and encrypting all IP packets. VPN uses a tunnel to encrypt all information at the IP level.