Technology Overview

Configuring Hierarchical VPLS

Release
13.1

Published: 2013-02-08

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Technology Overview Configuring Hierarchical VPLS
Release 13.1
NCE0015

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

Table of Contents

Introduction

This document provides detailed configuration guidance for configuring hierarchical virtual private LAN service (H-VPLS) using point-to-point pseudowires from spoke provider edge (PE) routers to hub PE routers.

VPLS Versions Overview

VPLS is one of the key MPLS-based services that have developed in the industry. The purpose of VPLS is to provide a private multipoint LAN-type Ethernet connectivity service. For those more familiar with technologies like Asynchronous Transfer Mode, VPLS is similar to a LAN emulation service for MPLS.

VPLS is especially useful in the service provider space as the way to deliver Layer 2 multipoint transparent services over an Ethernet infrastructure using MPLS. The key differentiating factor of VPLS is MPLS. There are different ways for a service provider to deliver services over an Ethernet infrastructure, but not all of them fit into the requirements that a service provider has in terms of scalability, reliability, service flexibility, and operational complexity. MPLS is the catalyst that can turn an Ethernet infrastructure into a carrier class network, making it suitable for a service provider. This is as opposed to a VLAN-based or Q-in-Q operation that does not provide what is required in the carrier environment.

VPLS, is the main technology in use in the Metro Ethernet space, with two standardized implementation options:

- RFC4761 – *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*
- RFC4762 – *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*

BGP-based VPLS and LDP-based VPLS are nearly identical in the operation of the forwarding plane, with the main differences in the control plane, particularly in the protocols used to signal and establish the pseudowires, BGP or LDP.

LDP-based VPLS Challenges

VPLS allows service providers to deploy carrier class services over an Ethernet-based network in a reliable and flexible way. Starting with business services and continuing with broadband multiplay services, service providers are gaining deployment experience with VPLS, and are also finding some of the challenges that this technology presents, especially in terms of scalability and interoperability.

LDP-based VPLS requires a full mesh of tunnel LSPs between all the PE routers that participate in the VPLS service. For each VPLS service, n*(n-1)/2 pseudowires must be set up between the PE routers. The full mesh requirement creates signaling overhead, consequently LDP-based VPLS has scaling challenges for large deployments.

LDP-signaled VPLS has the following issues:

- It is labor intensive because you must manually configure targeted LDP sessions.

- The requirement for a full mesh of pseudowires creates significant signaling overhead.
- Multicast, broadcast, and unknown unicast packets must be replicated for each provisioned pseudowire, which can waste bandwidth in large-scale deployments, especially for the hub router in a hub-and-spoke topology.

To address the scaling issues of LDP-based VPLS, hierarchical VPLS (H-VPLS) is defined in RFC 4762.

H-VPLS addresses two different issues:

- The signaling overhead caused by the requirement for a full mesh of pseudowires.
- The possibility of extending the VPLS domain to use simpler, less expensive devices.

Juniper Networks recommends using BGP-based VPLS for better scalability in the control plane and data plane. However, service providers are often in a situation where they need Juniper Networks routers to interoperate with other vendors' routers, which may not support BGP-based VPLS.

To support interoperability, Juniper Networks has two solutions:

1. Interworking between LDP-based VPLS and BGP-based VPLS on the border routers using *mesh groups*.
2. Configuring H-VPLS by terminating Martini pseudowires from the spoke PE routers to the hub VPLS PE routers using mesh groups.

For a detailed description of how H-VPLS is used, see *Demystifying H-VPLS*, at http://www.juniper.net/us/en/local/pdf/app-notes/3500116-en.pdf.

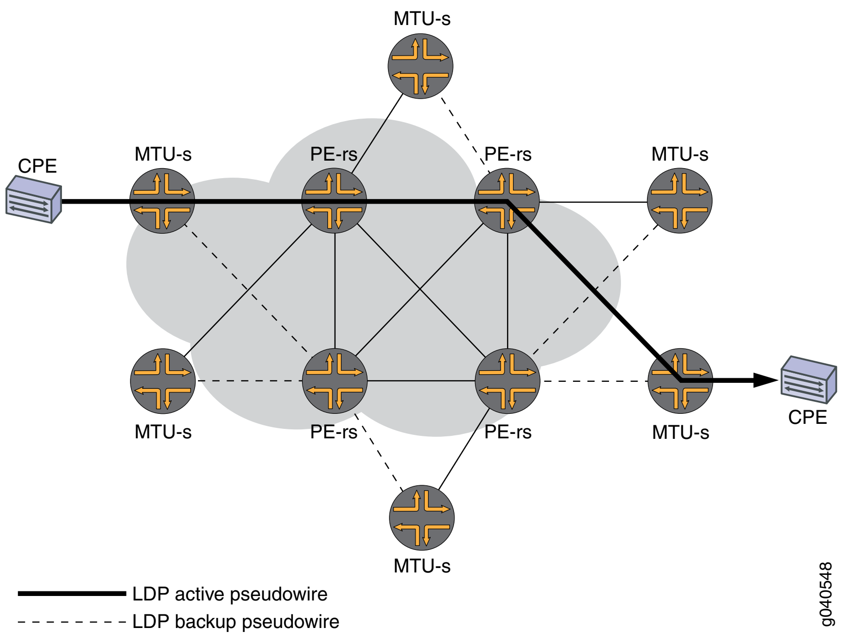
H-VPLS Implementation

Hierarchical LDP-based VPLS requires a full mesh of tunnel LSPs between all the PE routers that participate in the VPLS service. For each VPLS service, n*(n-1)/2 pseudowires must be set up between the PE routers. Although the full mesh requirement creates signaling overhead, the larger negative impact to large-scale deployment is the packet replication requirements for each provisioned pseudowire on a PE router. Using hierarchical connectivity reduces signaling and replication overhead to facilitate large-scale deployments.

H-VPLS defines the following new VPLS functions:

- PE-r (Hub-PE) — A PE router that has routing capabilities but does not have bridging capabilities. It supports all of the functions of the VPLS architecture. It has VPLS pseudowires to PE-rs routers and also has pseudowires with other devices called multi-tenant units (MTUs).
- PE-rs — A PE router that has routing and bridging (switching) capabilities.
- MTU-s (Spoke-PE) — A switch that has bridging capabilities but does not have routing capabilities. This represents the access layer of the H-VPLS architecture. The MTU-s device establishes pseudowires to one or two PE-rs routers through which VPLS traffic is forwarded.

Figure 1: Active and Backup Paths



H-VPLS Protocol Operation

The operation between PE-rs routers uses normal VPLS. Between MTU-s devices and PE-rs routers, the PE-rs routers treat the pseudowires as access links. Therefore, the split horizon rule used in normal VPLS is not used.

If traffic is received at a PE-rs router from an MTU-s device, it is forwarded to the other PE-rs routers and MTU-s devices that are connected to the same PE-rs router. When traffic is received at a PE-rs router from another PE-rs router, it is forwarded to the MTU-s devices connected to it through a pseudowire, but not to the other PE-rs routers. In this case the split horizon rule is used.

The mode of operation used by H-VPLS is intended to make VPLS more scalable. However, this mode of operation requires PE-rs routers to maintain media access control (MAC) tables and to perform the VPLS operations of learning and flooding. In normal VPLS, these routers are performing the role of provider (P) routers and have no VPLS state. In H-VPLS operation, a PE-rs router performs the VPLS operations of learning and flooding for all of the MTU-s devices to which it is connected. H-VPLS operation can lead to data plane scaling problems, especially in terms of the number and size of the MAC tables.

In summary, H-VPLS creates a control plane hierarchy, in the form of MTU-s devices and PE-rs routers, at the expense of forcing hierarchy in the data plane as well. Therefore, in the process of solving one scalability problem, H-VPLS introduces a new data plane scalability problem, and it does not provide solutions for this new problem.

It is important to note that the ability to extend the VPLS domain to less expensive and simpler devices by establishing pseudowires into a centralized or semi-centralized PE-rs router, is not an exclusive capability of LDP-based H-VPLS. This capability can be supported by BGP-based H-VPLS also.

Mesh Group Operation

Junos OS introduces the concept of a *mesh group*. A mesh group is used to connect multiple partial mesh domains into a single mesh group. Using a mesh group augments the forwarding plane operations to permit forwarding across mesh groups. A pseudowire mesh group is defined as a group of all pseudowires, that are fully meshed in the data plane. By default PE routers within the same mesh group do not communicate through the PE-r router .

The following are the H-VPLS definitions of flooding, learning, and learned unicast MAC forwarding:

- Flooding — Any broadcast, multicast, or unknown unicast packet received over a pseudowire and belonging to mesh group X must be forwarded to all the pseudowires of that instance, except those that are part of mesh group X.
- Learning — Source MAC address learning remains unchanged from normal VPLS.
- Learned unicast MAC forwarding — Any traffic received with a destination unicast MAC address learned on pseudowireX1 and belonging to mesh group X is forwarded only if the packet is received over a pseudowire that is not part of mesh group X.

To enable H-VPLS, configure an LDP Layer 2 circuit in a VPLS instance using mesh groups. The Layer 2 circuit virtual circuit ID must match the VPLS ID on the hub PE router's VPLS instance.

Junos OS supports up to 14 user-defined mesh groups per VPLS instance on MX series routers and up to 254 user-defined mesh groups per VPLS instance on M Series and T Series routers. In all cases, there are two default mesh groups created by the system.

Mesh Group Configuration Options

The following are descriptions of the two methods configuring H-VPLS using mesh groups:

1. Configure a mesh group for each Layer 2 circuit pseudowire terminating at a VPLS routing instance.
    - You can configure a maximum of 14 mesh groups on MX Series routers and a maximum of 254 mesh groups for M Series and T Series routers.
    - The **ethernet-ccc** encapsulation is used in one mesh group for each Layer 2 circuit configuration.
    - You can use different Layer 2 circuit and VPLS ID pairs for each spoke PE router mesh group.
    - You can terminate Layer 2 circuits into BGP-based VPLS or LDP-based VPLS on the hub PE router.
    - BGP-based VPLS is used in the configuration that uses one mesh group for each Layer 2 circuit.

2. Configure a single mesh group and terminate all the Layer 2 circuit pseudowires into it. Then enable local switching between the pseudowires by including the **local-switching** statement. The following applies to this method:

   - By default, local switching for mesh groups is not enabled. However, the **local-switching** statement is useful if you are:
     - Terminating Layer 2 circuit pseudowires from different spoke PE routers
     - Configuring the routers with the same virtual circuit ID and VPLS ID pairs in a mesh group
     - Configuring the routers for an LDP-signaled VPLS routing instance.
   - Layer 2 circuits can be terminated into BGP-based VPLS or LDP-based VPLS on the hub PE router.
   - LDP-based VPLS is used in the configuration that terminates all the Layer 2 circuit pseudowires into a single mesh group.

NOTE: **Pseudowire redundancy from spoke PE routers is supported if the MTU devices (spoke PE routers) are Juniper Networks routers, because pseudowire switchover is initiated by the spoke PE router in an H-VPLS scenario.**

Related Documentation

- Example: Configuring BGP-Based H-VPLS Using Different Mesh Groups for Each Spoke Router on page 5
- Example: Configuring LDP-Based H-VPLS Using a Single Mesh Group to Terminate the Layer 2 Circuits on page 28

Example: Configuring BGP-Based H-VPLS Using Different Mesh Groups for Each Spoke Router

This example shows how to configure the hierarchical virtual private LAN service (H-VPLS) using different mesh groups to provide H-VPLS functionality and provides steps for verifying the configuration. This is one type of H-VPLS configuration possible in the Juniper Networks implementation. For information about the alternate type of configuration see "Example: Configuring LDP-Based H-VPLS Using a Single Mesh Group to Terminate the Layer 2 Circuits" on page 28.

Using mesh groups improves LDP-based VPLS control plane scalability and avoids the requirement for a full mesh of LDP sessions. This example uses BGP-based VPLS.

This example is organized into the following sections:

Requirements

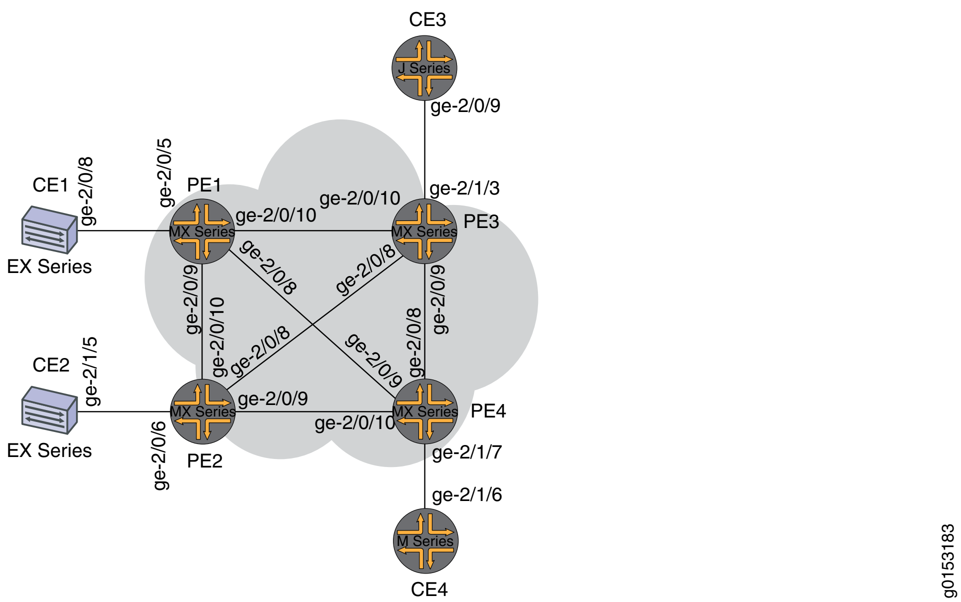
This example uses the following hardware components:

- Four MX Series 3D Universal Edge Routers for Router PE1, Router PE2, Router PE3, and Router PE4
- One M Series Multiservice Edge Router for Router CE4
- Two EX Series Ethernet Switches for Device CE1 and Device CE2
- One J Series Services Router for Router CE3

Overview and Topology

Figure 2 on page 6 shows the physical topology used in this example.

Figure 2: Physical Topology of H-VPLS



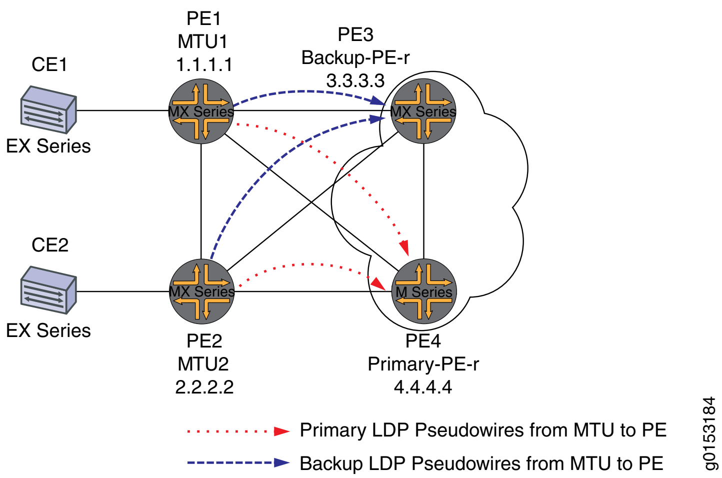
The following describes the base configuration used in this example:

- Router PE1 and Router PE2 are configured as MTU devices.
- Router PE3 and Router PE4 are configured as PE-r routers, each using an LDP-based VPLS routing instance.
- The LDP and OSPF protocols are configured on all of the MTU devices and PE-r routers.
- Core-facing interfaces are enabled with the MPLS address family.
- Optionally, the VPLS routing instances can be configured on PE-r routers with the **no-tunnel-interface** statement. This allows the routers to use a label-switched interface (LSI), which is useful if your routers do not have Tunnel Services PICs or built-in support for tunnel services.

- All of the routers are configured with loopback IP addresses.
- BGP is configured on the PE-r routers. Optionally, you can configure route reflection. This is useful for scaling internal BGP (IBGP). The BGP configuration includes the **signaling** statement at the **[edit protocols bgp group *group-name* family l2vpn]** hierarchy level to support Layer 2 VPN signaling using BGP.

Figure 3 on page 7 shows the logical topology used in this example.

Figure 3: Logical Topology of H-VPLS



In Figure 3 on page 7:

- The MTU devices (Router PE1 and Router PE2) have Layer 2 circuit connections to the PE-r routers (Router PE3 and Router PE4). For redundancy, a backup neighbor is configured for the Layer 2 circuit connections to the PE-r routers.
- The **l2circuit** statement in the **[edit protocols]** hierarchy is included on the MTU devices.
- A VPLS routing instance is configured on the PE-r routers.
- In the VPLS routing instance on the PE-r routers, mesh groups are created to terminate the Layer 2 circuit pseudowires that originate at the MTU devices.
- Each MTU device is configured with a different virtual circuit ID.
- Each PE-r router's mesh groups configuration includes VPLS ID values that match the virtual circuit IDs used on the MTU devices.

Configuration

To configure H-VPLS with different mesh groups for each spoke PE-r router using BGP-based VPLS, perform the following tasks:

- Configuring the Spoke MTU PE Routers on page 8
- Configuring the Hub PE (PE-r) on page 9
- Verifying the H-VPLS Operation on page 12

Configuring the Spoke MTU PE Routers

Step-by-Step Procedure

1. On Router PE1, configure the Gigabit Ethernet interface connected to Router CE1. Include the **encapsulation** statement and specify the **ethernet-ccc** option. Also configure the logical interface by including the **family** statement and specifying the **ccc** option.

```
[edit interfaces]
ge-2/0/5 {
  encapsulation ethernet-ccc;
  unit 0 {
    family ccc;
  }
}
```

2. On Router PE1, configure the Layer 2 circuit by including the **neighbor** statement and specifying the IP address of Router PE3 as the neighbor. Configure the Gigabit Ethernet logical interface by including the **virtual-circuit-id** statement and specifying **100** as the ID. Also configure a backup neighbor for the Layer 2 circuit by including the **backup-neighbor** statement, specifying the loopback interface IP address of Router PE4 as the backup neighbor, and including the **standby** statement.

```
[edit protocols]
l2circuit {
  neighbor 3.3.3.3 {
    interface ge-2/0/5.0 {
      virtual-circuit-id 100;
      backup-neighbor 4.4.4.4 { # Backup H-VPLS PE router
      standby;
    }
  }
}
```

3. On Router PE2, configure the Gigabit Ethernet interface connected to Router CE2. Include the **encapsulation** statement and specify the **ethernet-ccc** option. Also configure the logical interface by including the **family** statement and specifying the **ccc** option.

```
[edit interfaces]
ge-2/0/6 {
  encapsulation ethernet-ccc;
  unit 0 {
    family ccc;
  }
}
```

4. On Router PE2, configure the Layer 2 circuit by including the **neighbor** statement and specifying the IP address of Router PE3 as the neighbor. Configure the Gigabit Ethernet logical interface by including the **virtual-circuit-id** statement and specifying **200** as the ID. Configure the encapsulation by including the **encapsulation-type** statement and specifying the **ethernet** option. Also configure a backup neighbor for the Layer 2 circuit by including the **backup-neighbor** statement, specifying the loopback interface IP address of Router PE4 as the backup neighbor, and including the **standby** statement.

```
[edit protocols]
l2circuit {
   neighbor 3.3.3.3 {
      interface ge-1/0/2.0 {
         virtual-circuit-id 200; # different VC-ID
         encapsulation-type ethernet; # default encapsulation
         backup-neighbor 4.4.4.4 {
            standby;
         }
      }
   }
}
```

Configuring the Hub PE (PE-r)

Step-by-Step Procedure

1. On Router PE3 (the primary hub), configure the Gigabit Ethernet interface connected to Router CE3. Include the **encapsulation** statement and specify the **ethernet-vpls** option. Also configure the logical interface by including the **family vpls** statement.

```
[edit interfaces]
ge-2/0/0 {
   encapsulation ethernet-vpls;
   unit 0 {
      family vpls;
   }
}
lo0 {
   unit 0 {
      family inet {
         address 3.3.3.3/32;
      }
   }
}
```

2. On Router PE4 (the backup hub), configure the Gigabit Ethernet interface connected to Router CE4. Include the **encapsulation** statement and specify the **ethernet-vpls** option. Also configure the logical interface by including the **family vpls** statement.

```
[edit interfaces]
ge-2/1/7 {
   encapsulation ethernet-vpls;
   unit 0 {
      description to_CE4;
      family vpls;
   }
}

lo0 {
   unit 0 {
      family inet {
         address 4.4.4.4/32;
      }
   }
}
```

3. On PE-r Router PE3, configure the BGP-based VPLS routing instance by including the **instance-type** statement at the **[edit routing-instances H-VPLS]** hierarchy level and specifying the **vpls** option. Include the **interface** statement and specify the Gigabit Ethernet interface connected to Router CE3. Configure a route distinguisher to ensure that the route advertisement is unique by including the **route-distinguisher** statement and specifying **3.3.3.3:33** as the value. Also configure the VPN routing and forwarding (VRF) route target to be included in the route advertisements to the other routers participating in the VPLS. To configure the VRF route target, include the **vrf-target** statement and specify **target:64510:2** as the value. Optionally, include the **no-tunnel-services** statement to enable the use of LSI interfaces, which is useful if the device does not have tunnel services. The **no-tunnel-services** statement is omitted in this example. Optionally, you can include the **site-range** statement to specify an upper limit on the maximum site identifier that can be accepted to allow a pseudowire to be brought up. The **site-range** statement is omitted in this example. We recommend using the default of 65,534.

   Configure the VPLS protocol and the mesh groups for each MTU PE device.

   To configure the VPLS protocol, include the **vpls** statement at the **[edit routing-instances H-VPLS protocols]** hierarchy level. Include the **site** statement and specify a name for the site. Include the **interface** statement and specify the Gigabit Ethernet interface connected to Device CE3.

   Configuring mesh groups under the VPLS instance terminates the Layer 2 circuit into the VPLS instance. To configure each mesh group, include the **mesh-group** statement and specify the mesh group name. In this example, the mesh group name is the name of the MTU device associated with each mesh group. Include the **vpls-id** statement and specify the ID that matches the virtual circuit ID configured in "Configuring the Spoke MTU PE Routers" on page 8. Also include the **neighbor** statement and specify the IP address of the spoke PE router associated with each mesh group. Optionally, include the **local-switching** statement if you are not using a full mesh of VPLS connections. The **local-switching** statement is useful if you are configuring a single mesh group and terminating multiple Layer 2 circuit pseudowires into it. The **local-switching** statement is omitted in this example.

```
routing-instances {
  H-VPLS {
    instance-type vpls;
    interface ge-2/1/3.0;
    route-distinguisher 3.3.3.3:33;
    vrf-target target:64510:2;
    protocols {
      vpls {
        site pe3 {
          site-identifier 3;
          interface ge-2/1/3.0;
        }
        mesh-group pe1 {
          vpls-id 100;
          neighbor 1.1.1.1;
        }
        mesh-group pe2 {
          vpls-id 200;
```



```
                neighbor 2.2.2.2;
              }
            }
          }
        }
      }
```

4. On PE-r Router PE4, configure a routing instance like the one on Router PE3.

```
routing-instances {
  H-VPLS {
    instance-type vpls;
    interface ge-2/1/7.0;
    route-distinguisher 4.4.4.4:44;
    vrf-target target:64510:2;
    protocols {
      vpls {
        site pe4 {
          site-identifier 4;
          interface ge-2/1/7.0;
        }
        mesh-group pe1 {
          vpls-id 100;
          neighbor 1.1.1.1;
        }
        mesh-group pe2 {
          vpls-id 200;
          neighbor 2.2.2.2;
        }
      }
    }
  }
}
```

Verifying the H-VPLS Operation

Step-by-Step Procedure

This section describes the operational commands that you can use to validate that the H-VPLS is working as expected.

1. On Router PE1 and Router PE2, use the **show l2circuit connections** command to verify that the Layer 2 circuit to Router PE3 is **Up** and the Layer 2 circuit to Router PE4 is in **standby** mode.

   The output also shows the assigned label, virtual circuit ID, and the **ETHERNET** encapsulation type.

```
user@PE1> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 3.3.3.3
    Interface                 Type  St     Time last up          # Up trans

    ge-2/0/5.0(vc 100)        rmt   Up     Oct 18 15:55:07 2012           1

      Remote PE: 3.3.3.3, Negotiated control-word: No
      Incoming label: 299840, Outgoing label: 800001
      Negotiated PW status TLV: No
      Local interface: ge-2/0/5.0, Status: Up, Encapsulation: ETHERNET
Neighbor: 4.4.4.4
    Interface                 Type  St     Time last up          # Up trans

    ge-2/0/5.0(vc 100)        rmt   ST
```

```
user@PE2> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 3.3.3.3
    Interface                 Type  St     Time last up          # Up trans
```

```
    ge-2/0/6.0(vc 200)        rmt   Up     Oct 18 15:55:07 2012          1

      Remote PE: 3.3.3.3, Negotiated control-word: No
      Incoming label: 299872, Outgoing label: 800002
      Negotiated PW status TLV: No
      Local interface: ge-2/0/6.0, Status: Up, Encapsulation: ETHERNET
Neighbor: 4.4.4.4
    Interface                 Type  St     Time last up          # Up trans

    ge-2/0/6.0(vc 200)        rmt   ST
```

2. On Router PE1 and Router PE2, use the **show ldp neighbor** command to verify that the targeted LDP sessions have been created between the loopback interface to the primary and backup H-VPLS hub neighbors.

```
user@PE1> show ldp neighbor
Address            Interface          Label space ID         Hold time
10.10.3.2          ge-2/0/9.0        2.2.2.2:0                13
10.10.1.2          ge-2/0/10.0        3.3.3.3:0                10
3.3.3.3            lo0.0              3.3.3.3:0                36
4.4.4.4            lo0.0              4.4.4.4:0                39
10.10.9.2          ge-2/0/8.0       4.4.4.4:0               14
```

```
user@PE2> show ldp neighbor
Address            Interface          Label space ID         Hold time
10.10.3.1          ge-2/0/10.0        1.1.1.1:0                12
10.10.5.2          ge-2/0/9.0        4.4.4.4:0                11
10.10.4.1          ge-2/0/8.0       3.3.3.3:0               11
3.3.3.3            lo0.0              3.3.3.3:0                39
4.4.4.4            lo0.0              4.4.4.4:0                38
```

3. On Router PE3 and Router PE4, use the **show vpls connections** command to verify that the VPLS connection status is **Up** for both the LDP-based VPLS and the BGP-based VPLS Layer 2 circuits that are terminated.

```
user@PE3> show vpls connections

Layer-2 VPN connections:

Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not
CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not
same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not available
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy
RS -- remote site standby        SN -- Static Neighbor
LB -- Local site not best-site   RB -- Remote site not best-site
VM -- VLAN ID mismatch
```



```
Legend for interface status
Up -- operational
Dn -- down

Instance: H-VPLS
  BGP-VPLS State
  Local site: pe3 (3)
    connection-site           Type  St     Time last up          # Up trans

    4                         rmt   Up     Oct 18 15:58:39 2012           1

      Remote PE: 4.4.4.4, Negotiated control-word: No
      Incoming label: 800267, Outgoing label: 800266
      Local interface: vt-2/0/9.135266562, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls H-VPLS local site 3 remote site 4
  LDP-VPLS State
  Mesh-group connections: pe1
    Neighbor                  Type  St     Time last up          # Up trans

    1.1.1.1(vpls-id 100)      rmt   Up     Oct 18 15:55:07 2012           1

      Remote PE: 1.1.1.1, Negotiated control-word: No
      Incoming label: 800001, Outgoing label: 299840
      Negotiated PW status TLV: No
      Local interface: vt-2/0/10.135266560, Status: Up, Encapsulation:
ETHERNET
        Description: Intf - vpls H-VPLS neighbor 1.1.1.1 vpls-id 100
  Mesh-group connections: pe2
    Neighbor                  Type  St     Time last up          # Up trans

    2.2.2.2(vpls-id 200)      rmt   Up     Oct 18 15:55:07 2012           1

      Remote PE: 2.2.2.2, Negotiated control-word: No
      Incoming label: 800002, Outgoing label: 299872
      Negotiated PW status TLV: No
     Local interface: vt-2/0/8.135266561, Status: Up, Encapsulation: ETHERNET

        Description: Intf - vpls H-VPLS neighbor 2.2.2.2 vpls-id 200
```

```
user@PE4> show vpls connections

Layer-2 VPN connections:

Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not
CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not
same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not available
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy
```

```
RS -- remote site standby        SN -- Static Neighbor
LB -- Local site not best-site   RB -- Remote site not best-site
VM -- VLAN ID mismatch

Legend for interface status
Up -- operational
Dn -- down

Instance: H-VPLS
  BGP-VPLS State
  Local site: pe4 (4)
    connection-site           Type  St     Time last up          # Up trans

    3                         rmt   Up     Oct 18 15:58:39 2012           1

      Remote PE: 3.3.3.3, Negotiated control-word: No
      Incoming label: 800266, Outgoing label: 800267
      Local interface: vt-2/0/8.17826050, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls H-VPLS local site 4 remote site 3
  LDP-VPLS State
  Mesh-group connections: pe1
    Neighbor                  Type  St     Time last up          # Up trans

    1.1.1.1(vpls-id 100)      rmt   Up     Oct 18 15:58:39 2012           1

      Remote PE: 1.1.1.1, Negotiated control-word: No
      Incoming label: 800002, Outgoing label: 299856
      Negotiated PW status TLV: No
     Local interface: vt-2/0/9.17826048, Status: Up, Encapsulation: ETHERNET

        Description: Intf - vpls H-VPLS neighbor 1.1.1.1 vpls-id 100
  Mesh-group connections: pe2
    Neighbor                  Type  St     Time last up          # Up trans

    2.2.2.2(vpls-id 200)      rmt   Up     Oct 18 15:58:39 2012           1

      Remote PE: 2.2.2.2, Negotiated control-word: No
      Incoming label: 800003, Outgoing label: 299888
      Negotiated PW status TLV: No
     Local interface: vt-2/0/10.17826049, Status: Up, Encapsulation: ETHERNET

        Description: Intf - vpls H-VPLS neighbor 2.2.2.2 vpls-id 200
```

4. On Router PE3 and Router PE4, use the **show vpls flood** command to verify that the H-VPLS PE router created a flood group for each spoke PE site.

```
user@PE3> show vpls flood
Name: H-VPLS
CEs: 1
VEs: 3
Flood Routes:
  Prefix    Type           Owner                 NhType          NhIndex
  0x300cc/51 FLOOD_GRP_COMP_NH __ves__           comp            1376
  0x300cf/51 FLOOD_GRP_COMP_NH __all_ces__       comp            744
  0x300d5/51 FLOOD_GRP_COMP_NH pe1               comp            1702
  0x300d3/51 FLOOD_GRP_COMP_NH pe2               comp            1544
  0x30001/51 FLOOD_GRP_COMP_NH __re_flood__      comp            740
```

```
user@PE4> show vpls flood
Name: H-VPLS
CEs: 1
VEs: 3
```



```
Flood Routes:
  Prefix    Type           Owner                 NhType          NhIndex
  0x300d1/51 FLOOD_GRP_COMP_NH __ves__           comp            1534
  0x300d0/51 FLOOD_GRP_COMP_NH __all_ces__       comp            753
  0x300d6/51 FLOOD_GRP_COMP_NH pe1               comp            1378
  0x300d4/51 FLOOD_GRP_COMP_NH pe2               comp            1695
  0x30002/51 FLOOD_GRP_COMP_NH __re_flood__      comp            750
```

5. On Router PE3 and Router PE4, use the **show vpls mac-table** command to verify that MAC addresses of the CE devices have been learned.

```
user@PE3> show vpls mac-table
MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC

          SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : H-VPLS
 Bridging domain : __H-VPLS__, VLAN : NA
   MAC                 MAC      Logical          NH     RTR
   address             flags    interface        Index  ID
   00:21:59:0f:35:32   D        vt-2/0/8.135266560
   00:21:59:0f:35:33   D        ge-2/1/3.0
   00:21:59:0f:35:d4   D        vt-2/0/9.135266561
   00:21:59:0f:35:d5   D        vt-2/0/10.135266562
```

```
user@PE4> show vpls mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC

          SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Logical system   : PE4
Routing instance : H-VPLS
 Bridging domain : __H-VPLS__, VLAN : NA
   MAC                 MAC      Logical          NH     RTR
   address             flags    interface        Index  ID
   00:21:59:0f:35:32   D        vt-2/0/8.17826050
   00:21:59:0f:35:33   D        vt-2/0/9.17826050
   00:21:59:0f:35:d4   D        vt-2/0/10.17826050
   00:21:59:0f:35:d5   D        ge-2/1/7.0
```

6. Make sure that the CE devices can ping each other.

```
user@CE1> ping 10.255.14.219 # ping sent from CE1 CE4
PING 10.255.14.219 (10.255.14.219): 56 data bytes
64 bytes from 10.255.14.219: icmp_seq=0 ttl=64 time=10.617 ms
64 bytes from 10.255.14.219: icmp_seq=1 ttl=64 time=9.224 ms
^C
--- 10.255.14.219 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 9.224/9.921/10.617/0.697 ms
```

```
user@CE2> ping 10.255.14.218 # ping sent from CE2 to CE3

PING 10.255.14.218 (10.255.14.218): 56 data bytes
64 bytes from 10.255.14.218: icmp_seq=0 ttl=64 time=1.151 ms
64 bytes from 10.255.14.218: icmp_seq=1 ttl=64 time=0.674 ms
^C
--- 10.255.14.218 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.674/0.913/1.151/0.238 ms
```

7. Check the relevant routing tables.

```
user@PE1> show route table l2circuit.0
l2circuit.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

3.3.3.3:NoCtrlWord:5:100:Local/96
                   *[L2CKT/7] 00:12:16, metric2 1
                    > to 10.10.1.2 via ge-2/0/10.0
3.3.3.3:NoCtrlWord:5:100:Remote/96
                   *[LDP/9] 00:12:16
                      Discard
4.4.4.4:NoCtrlWord:5:100:Local/96
                   *[L2CKT/7] 00:12:10, metric2 1
                    > to 10.10.9.2 via ge-2/0/8.0
4.4.4.4:NoCtrlWord:5:100:Remote/96
                   *[LDP/9] 00:12:15
                      Discard
```

```
user@PE2> show route table l2circuit.0
l2circuit.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

3.3.3.3:NoCtrlWord:5:200:Local/96
                   *[L2CKT/7] 00:13:13, metric2 1
                    > to 10.10.4.1 via ge-2/0/8.0
3.3.3.3:NoCtrlWord:5:200:Remote/96
                   *[LDP/9] 00:13:13
                      Discard
4.4.4.4:NoCtrlWord:5:200:Local/96
                   *[L2CKT/7] 00:13:13, metric2 1
                    > to 10.10.5.2 via ge-2/0/9.0
4.4.4.4:NoCtrlWord:5:200:Remote/96
                   *[LDP/9] 00:13:13
                      Discard
```

```
user@PE3> show route table H-VPLS.l2vpn.0

H-VPLS.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

3.3.3.3:33:3:1/96
                   *[L2VPN/170/-101] 03:19:26, metric2 1
                      Indirect
4.4.4.4:44:4:1/96
                   *[BGP/170] 03:15:45, localpref 100, from 4.4.4.4
                      AS path: I, validation-state: unverified
                    > to 10.10.6.2 via ge-2/0/9.0
```

```
user@PE4> show route table H-VPLS.l2vpn.0

H-VPLS.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

3.3.3.3:33:3:1/96
                   *[BGP/170] 03:21:17, localpref 100, from 3.3.3.3
                      AS path: I, validation-state: unverified
                    > to 10.10.6.1 via ge-2/0/9.0
4.4.4.4:44:4:1/96
                   *[L2VPN/170/-101] 03:17:47, metric2 1
                      Indirect
```


Results The configuration and verification parts of this example have been completed. The following section is for your reference.

The relevant sample configuration for Router PE1 follows.

Router PE1

```
interfaces {
  ge-2/0/5 {
    encapsulation ethernet-ccc;
    unit 0 {
      description to_CE1;
      family ccc;
    }
  }
  ge-2/0/8 {
    unit 0 {
      description to_PE4;
      family inet {
        address 10.10.9.1/30;
      }
      family mpls;
    }
  }
  ge-2/0/9 {
    unit 0 {
      description to_PE2;
      family inet {
        address 10.10.3.1/30;
      }
      family mpls;
    }
  }
  ge-2/0/10 {
    unit 0 {
      description to_PE3;
      family inet {
        address 10.10.1.1/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 1.1.1.1/32;
      }
    }
  }
}
protocols {
  mpls {
    interface ge-2/0/8.0;
    interface ge-2/0/9.0;
    interface ge-2/0/10.0;
  }
  ospf {
    traffic-engineering;
```

```
        area 0.0.0.0 {
          interface lo0.0 {
            passive;
          }
          interface ge-2/0/8.0;
          interface ge-2/0/9.0;
          interface ge-2/0/10.0;
        }
      }
      ldp {
        interface ge-2/0/8.0;
        interface ge-2/0/9.0;
        interface ge-2/0/10.0;
        interface lo0.0;
      }
      l2circuit {
        neighbor 3.3.3.3 {
          interface ge-2/0/5.0 {
            virtual-circuit-id 100;
            backup-neighbor 4.4.4.4 {
              standby;
            }
          }
        }
      }
    }
```

The relevant sample configuration for Router PE2 follows.

Router PE2

```
interfaces {
  ge-2/0/6 {
    encapsulation ethernet-ccc;
    unit 0 {
      description to_CE2;
      family ccc;
    }
  }
  ge-2/0/8 {
    unit 0 {
      description to_PE3;
      family inet {
        address 10.10.4.2/30;
      }
      family mpls;
    }
  }
  ge-2/0/9 {
    unit 0 {
      description to_PE4;
      family inet {
        address 10.10.5.1/30;
      }
      family mpls;
    }
  }
  ge-2/0/10 {
```

```
        unit 0 {
            description to_PE1;
            family inet {
                address 10.10.3.2/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 2.2.2.2/32;
            }
        }
    }
}
protocols {
    mpls {
        interface ge-2/0/8.0;
        interface ge-2/0/9.0;
        interface ge-2/0/10.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface ge-2/0/8.0;
            interface ge-2/0/9.0;
            interface ge-2/0/10.0;
        }
    }
    ldp {
        interface ge-2/0/8.0;
        interface ge-2/0/9.0;
        interface ge-2/0/10.0;
        interface lo0.0;
    }
    l2circuit {
        neighbor 3.3.3.3 {
            interface ge-2/0/6.0 {
                virtual-circuit-id 200;
                backup-neighbor 4.4.4.4 {
                    standby;
                }
            }
        }
    }
}
```

The relevant sample configuration for Router PE3 follows.

Router PE3

```
interfaces {
    ge-2/0/8 {
        unit 0 {
```

```
            description to_PE2;
            family inet {
                address 10.10.4.1/30;
            }
            family mpls;
        }
    }
    ge-2/0/9 {
        unit 0 {
            description to_PE4;
            family inet {
                address 10.10.6.1/30;
            }
            family mpls;
        }
    }
    ge-2/0/10 {
        unit 0 {
            description to_PE1;
            family inet {
                address 10.10.1.2/30;
            }
            family mpls;
        }
    }
    ge-2/1/3 {
        encapsulation ethernet-vpls;
        unit 0 {
            description to_CE3;
            family vpls;
        }
    }
    lo0 {
        unit 0{
            family inet {
                address 3.3.3.3/32;
            }
        }
    }
}
protocols {
    mpls {
        interface ge-2/0/8.0;
        interface ge-2/0/9.0;
        interface ge-2/0/10.0;
    }
    bgp {
        group internal-peers {
            type internal;
            local-address 3.3.3.3;
            family l2vpn {
                signaling;
            }
            neighbor 4.4.4.4;
        }
    }
```



```
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface ge-2/0/8.0;
            interface ge-2/0/9.0;
            interface ge-2/0/10.0;
        }
    }
    ldp {
        interface ge-2/0/8.0;
        interface ge-2/0/9.0;
        interface ge-2/0/10.0;
        interface lo0.0;
    }
}
routing-instances {
    H-VPLS {
        instance-type vpls;
        interface ge-2/1/3.0;
        route-distinguisher 3.3.3.3:33;
        vrf-target target:64510:2;
        protocols {
            vpls {
                site pe3 {
                    site-identifier 3;
                    interface ge-2/1/3.0;
                }
                mesh-group pe1 {
                    vpls-id 100;
                    neighbor 1.1.1.1;
                }
                mesh-group pe2 {
                    vpls-id 200;
                    neighbor 2.2.2.2;
                }
            }
        }
    }
}
routing-options {
    autonomous-system 64510;
}
```

The relevant sample configuration for Router PE4 follows.

Router PE4

```
interfaces {
    ge-2/0/8 {
        unit 0 {
            description to_PE3;
            family inet {
                address 10.10.6.2/30;
            }
            family mpls;
```

```
            }
        }
        ge-2/0/9 {
            unit 0 {
                description to_PE1;
                family inet {
                    address 10.10.9.2/30;
                }
                family mpls;
            }
        }
        ge-2/0/10 {
            unit 0 {
                description to_PE2;
                family inet {
                    address 10.10.5.2/30;
                }
                family mpls;
            }
        }
        ge-2/1/7 {
            encapsulation ethernet-vpls;
            unit 0 {
                description to_CE4;
                family vpls;
            }
        }
        lo0 {
            unit 0 {
                family inet {
                    address 4.4.4.4/32;
                }
            }
        }
    }
    protocols {
        mpls {
            interface ge-2/0/8.0;
            interface ge-2/0/9.0;
            interface ge-2/0/10.0;
        }
        bgp {
            group internal-peers {
                type internal;
                local-address 4.4.4.4;
                family l2vpn {
                    signaling;
                }
                neighbor 3.3.3.3;
            }
        }
        ospf {
            traffic-engineering;
            area 0.0.0.0 {
                interface lo0.0 {
                    passive;
```

```
        }
        interface ge-2/0/8.0;
        interface ge-2/0/9.0;
        interface ge-2/0/10.0;
      }
    }
    ldp {
      interface ge-2/0/8.0;
      interface ge-2/0/9.0;
      interface ge-2/0/10.0;
      interface lo0.0;
    }
  }
  routing-instances {
    H-VPLS {
      instance-type vpls;
      interface ge-2/1/7.0;
      route-distinguisher 4.4.4.4:44;
      vrf-target target:64510:2;
      protocols {
        vpls {
          site pe4 {
            site-identifier 4;
            interface ge-2/1/7.0;
          }
          mesh-group pe1 {
            vpls-id 100;
            neighbor 1.1.1.1;
          }
          mesh-group pe2 {
            vpls-id 200;
            neighbor 2.2.2.2;
          }
        }
      }
    }
  }
  routing-options {
    autonomous-system 64510;
  }
```

The relevant sample configuration for Device CE1 follows.

Router CE1

```
interfaces {
  ge-2/0/8 {
    unit 0 {
      description to_PE1;
      family inet {
        address 172.16.0.1/24;
      }
    }
  }
  lo0 {
    unit 0{
      family inet {
        address 10.255.14.214/32;
```

```
            }
          }
        }
      }
      protocols {
        ospf {
          area 0.0.0.0 {
            interface lo0.0 {
              passive;
            }
            interface ge-2/0/8.0;
          }
        }
      }
```

The relevant sample configuration for Device CE2 follows.

Router CE2

```
      interfaces {
        ge-2/1/5 {
          unit 0 {
            description to_PE2;
            family inet {
              address 172.16.0.2/24;
            }
          }
        }
        lo0 {
          unit 0{
            family inet {
              address 10.255.14.215/32;
            }
          }
        }
      }
      protocols {
        ospf {
          area 0.0.0.0 {
            interface lo0.0 {
              passive;
            }
            interface ge-2/1/5.0;
          }
        }
      }
```

The relevant sample configuration for Device CE3 follows.

Router CE3

```
      interfaces {
        ge-2/0/9 {
          unit 0 {
            description to_PE3;
            family inet {
              address 172.16.0.3/24;
            }
          }
        }
        lo0 {
```



```
    unit 0 {
      family inet {
        address 10.255.14.218/32;
      }
    }
  }
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface ge-2/0/9.0;
    }
  }
}
```

The relevant sample configuration for Device CE4 follows.

Router CE4

```
interfaces {
  ge-2/1/6 {
    unit 0 {
      description to_PE4;
      family inet {
        address 172.16.0.4/24;
      }
    }
  }
  lo0 {
    unit 0{
      family inet {
        address 10.255.14.219/32;
      }
    }
  }
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface ge-2/1/6.0;
    }
  }
}
```

Related Documentation

- Example: Configuring LDP-Based H-VPLS Using a Single Mesh Group to Terminate the Layer 2 Circuits on page 28
- VPLS Versions Overview on page 1

Example: Configuring LDP-Based H-VPLS Using a Single Mesh Group to Terminate the Layer 2 Circuits

This example shows how to configure a single mesh group to terminate the Layer 2 circuits into an LDP-based VPLS. This is one type of hierarchical virtual private LAN service (H-VPLS) configuration possible in the Juniper Networks implementation. For information about the alternate type of configuration see "Example: Configuring BGP-Based H-VPLS Using Different Mesh Groups for Each Spoke Router" on page 5.

This example provides step-by-step configuration instructions and also provides steps for verifying and troubleshooting the configuration.

This example is organized into the following sections:

Requirements

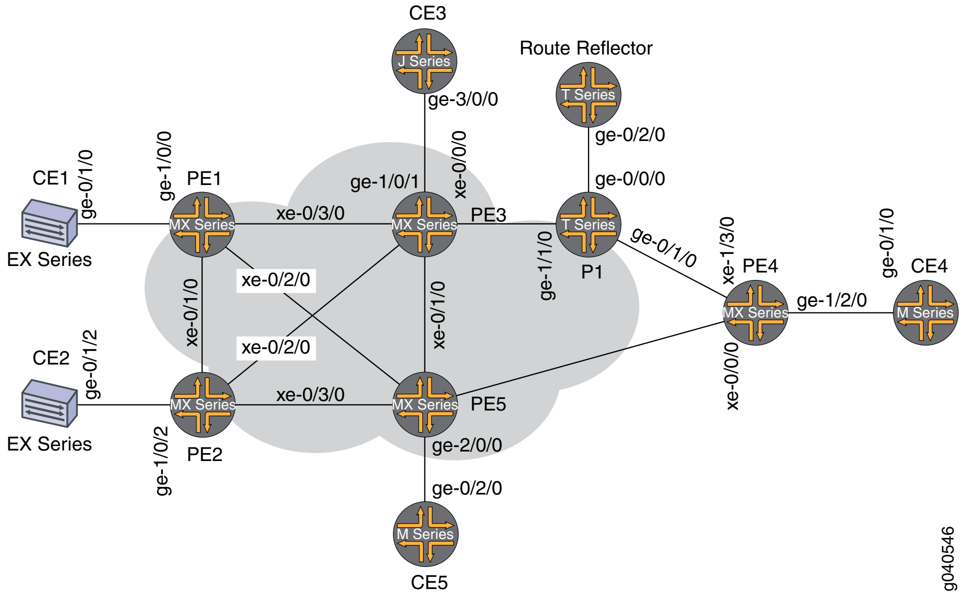
This example uses the following hardware components:

- Four MX Series 3D Universal Edge Routers for Routers PE1, PE2, PE3, and PE4
- Two M Series Multiservice Edge Routers for Routers CE4 and PE5
- Two EX Series Ethernet Switches for Devices CE1 and CE2
- Two T Series Core Routers for Routers P1 and the route reflector
- One J Series Services Router for Router CE3

Overview and Topology

Figure 4 on page 29 shows the physical topology used in this example.

Figure 4: Physical Topology of H-VPLS using a Single Mesh Group



In Figure 4 on page 29:

- Local switching is used to switch traffic between Layer 2 circuit pseudowires from the different spoke PE routers.
- The spoke PE routers are configured with the same virtual circuit ID and VPLS ID pair in a mesh group.
- The spoke PE routers are configured in an LDP-signaled VPLS routing instance.
- The layer 2 circuits are terminated into the LDP-based VPLS.

Configuration

To configure a single mesh group to terminate the Layer 2 circuits into an LDP-based VPLS, perform the following tasks:

- Configuring the Spoke PE Routers on page 29
- Configuring the Hub PE Router on page 31
- Verification on page 32

Configuring the Spoke PE Routers

Step-by-Step Procedure

Configure a single mesh group to terminate all the Layer 2 circuit pseudowires and enable local switching between the pseudowires.

1. On Router PE1, configure the Layer 2 circuit by including the **l2circuit** statement at the **[edit protocols]** hierarchy level. Include the **neighbor** statement and specify the IPv4 address of the hub PE router. Also configure the logical interface by including the **interface** statement and specify the interface connected to Router CE1.

Configure the virtual circuit ID by including the **virtual-circuit-id** statement and specifying **100** as the ID value at the **[edit protocols l2circuit neighbor 5.5.5.5 interface ge-1/0/0.0]** hierarchy level.

Configure the backup neighbor by including the **backup-neighbor** statement and specifying the IPv4 address of the backup hub PE router. Router PE3 is the backup neighbor in this example. Also include the **standby** statement at the **[edit protocols l2circuit neighbor 5.5.5.5 interface ge-1/0/0.0 backup-neighbor 3.3.3.3]** hierarchy level.

```
[edit protocols]
l2circuit {
   neighbor 5.5.5.5 {
      interface ge-1/0/0.0 {
         virtual-circuit-id 100;
         backup-neighbor 3.3.3.3 {
            standby;
         }
      }
   }
}
```

2. On Router PE2, configure the Layer 2 circuit by including the **l2circuit** statement at the **[edit protocols]** hierarchy level. Include the **neighbor** statement and specify the IPv4 address of the hub PE router. Configure the logical interface by including the **interface** statement and specifying the interface connected to Router CE2.

   Configure the virtual circuit ID by including the **virtual-circuit-id** statement and specifying **100** as the ID value at the **[edit protocols l2circuit neighbor 5.5.5.5 interface ge-1/0/2.0]** hierarchy level. Include the **encapsulation** statement and specify **ethernet** as the type.

   Configure the backup neighbor by including the **backup-neighbor** statement and specifying the IPv4 address of the backup hub PE router. Router PE3 is the backup neighbor in this example. Also include the **standby** statement at the **[edit protocols l2circuit neighbor 5.5.5.5 interface ge-1/0/0.0 backup-neighbor 3.3.3.3]** hierarchy level.

```
[edit protocols]
l2circuit {
   neighbor 5.5.5.5 {
      interface ge-1/0/2.0 {
         virtual-circuit-id 100;
         encapsulation-type ethernet;
         backup-neighbor 3.3.3.3 {
            standby;
         }
      }
   }
}
```

3. On Router PE4, configure the Layer 2 circuit by including the **l2circuit** statement at the **[edit protocols]** hierarchy level. Include the **neighbor** statement and specify the IPv4 address of the hub PE router. Configure the logical interface by including the **interface** statement and specify the interface connected to Router CE4.

Configure the virtual circuit ID by including the **virtual-circuit-id** statement and specifying **100** as the ID value at the **[edit protocols l2circuit neighbor 5.5.5.5 interface ge-1/2/0.0]** hierarchy level.

Configure the backup neighbor by including the **backup-neighbor** statement and specifying the IPv4 address of the backup hub PE router. Router PE3 is the backup neighbor in this example. Also include the **standby** statement at the **[edit protocols l2circuit neighbor 5.5.5.5 interface ge-1/2/0.0 backup-neighbor 3.3.3.3]** hierarchy level.

```
[edit protocols]
l2circuit {
   neighbor 5.5.5.5 {
      interface ge-1/2/0.0 {
         virtual-circuit-id 100;
         backup-neighbor 3.3.3.3 {
            standby;
         }
      }
   }
}
```

Configuring the Hub PE Router

Step-by-Step Procedure

Configure a single mesh group to terminate all the Layer 2 circuit pseudowires and enable local switching between the pseudowires.

1. On Router PE3, configure the Gigabit Ethernet interface connected to Router CE3 by including the **encapsulation** statement and specifying the **ethernet-vpls** option. Also configure the logical interface by including the **family** statement and specifying the **vpls** option.

```
[edit interfaces]
ge-1/0/1 {
   encapsulation ethernet-vpls;
   unit 0 {
      family vpls;
   }
}
```

2. On Router PE3, configure the logical loopback interface by including the **family** statement and specifying the **inet** option. Include the **address** statement and specify the IPv4 address for the interface.

```
[edit interfaces]
lo0 {
   unit 0 {
      family inet {
         address 3.3.3.3/32;
      }
   }
}
```

3. On Router PE3, configure the LDP-based VPLS routing instance by including the **instance-type** statement at the **[edit routing-instances H-VPLS]** hierarchy level and

specifying the **vpls** option. Include the **interface** statement and specify the Gigabit Ethernet interface connected to Router CE3.

Configure the VPLS protocol by including the **vpls** statement at the **[edit routing-instances H-VPLS protocols]** hierarchy level. Include the **no-tunnel-services** statement to enable the router to use an LSI interface.

```
[edit routing-instances]
H-VPLS {
  instance-type vpls;
  interface ge-1/0/1.0;
  protocols {
    vpls {
      no-tunnel-services;
    }
  }
}
```

4. On Router PE3, configure the mesh group by including the **mesh-group** statement at the **[edit routing-instances H-VPLS protocols vpls]** hierarchy level and specifying **L2-Circuits** as the name of the group. Include the **vpls-id** statement and specify **100** as the ID value. Include the **local-switching** statement to enable the router to switch traffic between the pseudowires.

   For each neighbor in the mesh group, include the **neighbor** statement and specify the IPv4 address of the spoke PE router.

```
[edit routing-instances H-VPLS protocols vpls]
mesh-group L2-Circuits {
  vpls-id 100;   <<< Same VPLS ID on all MTUs
  local-switching;   << Local-switching enabled
  neighbor 1.1.1.1;   <<MTU IP addresses
  neighbor 2.2.2.2;
  neighbor 4.4.4.4;
}
```

Verification

Step-by-Step Procedure

1. On Router PE5, use the **show ldp neighbor** command to verify that LDP sessions have been created to each of the spoke PE routers.

```
user@PE5# show ldp neighbor
Address            Interface          Label space ID          Hold time
1.1.1.1            lo0.0              1.1.1.1:0                 33
2.2.2.2            lo0.0              2.2.2.2:0                 37
4.4.4.4            lo0.0              4.4.4.4:0                 39
```

2. On Router PE5, use the **show vpls connections extensive** command to verify that the mesh group neighbor session is **Up**, that inbound and outbound labels have been assigned, that the VPLS ID is correct, and that the virtual tunnel interface is being used.

```
user@PE5# show vpls connections extensive
...
Instance: H-VPLS
    Number of local interfaces: 1
    Number of local interfaces up: 1
```

```
  Number of VE mesh-groups: 2
  Number of VE mesh-groups up: 1
  ge-2/0/0.0
  Mesh-group interfaces: L2-Circuits
    State: Up      ID: 2
    vt-2/1/0.1048848    Intf - vpls H-VPLS neighbor 4.4.4.4 vpls-id 100
    vt-2/1/0.1048849    Intf - vpls H-VPLS neighbor 2.2.2.2 vpls-id 100
    vt-2/1/0.1048850    Intf - vpls H-VPLS neighbor 1.1.1.1 vpls-id 100
  Mesh-group interfaces: __ves__
    State: Dn      ID: 0
Mesh-group connections: L2-Circuits
  Neighbor                  Type  St     Time last up          # Up trans

  4.4.4.4(vpls-id 100)      rmt   Up     Jan  3 16:46:26 2010           1

    Remote PE: 4.4.4.4, Negotiated control-word: No
    Incoming label: 800011, Outgoing label: 301088
    Local interface: vt-2/1/0.1048848, Status: Up, Encapsulation: ETHERNET

      Description: Intf - vpls H-VPLS neighbor 4.4.4.4 vpls-id 100
  Connection History:
      Jan  3 16:46:26 2010  status update timer
      Jan  3 16:46:26 2010  PE route changed
      Jan  3 16:46:26 2010  In lbl Update                     800011
      Jan  3 16:46:26 2010  Out lbl Update                    301088
      Jan  3 16:46:26 2010  In lbl Update                     800011
      Jan  3 16:46:26 2010  loc intf up             vt-2/1/0.1048848
  2.2.2.2(vpls-id 100)      rmt   Up     Jan  3 16:46:26 2010           1

    Remote PE: 2.2.2.2, Negotiated control-word: No
    Incoming label: 800010, Outgoing label: 301488
   Local interface: vt-2/1/0.1048849, Status: Up, Encapsulation: ETHERNET

      Description: Intf - vpls H-VPLS neighbor 2.2.2.2 vpls-id 100
  Connection History:
      Jan  3 16:46:26 2010  status update timer
      Jan  3 16:46:26 2010  PE route changed
      Jan  3 16:46:26 2010  In lbl Update                     800010
      Jan  3 16:46:26 2010  Out lbl Update                    301488
      Jan  3 16:46:26 2010  In lbl Update                     800010
      Jan  3 16:46:26 2010  loc intf up             vt-2/1/0.1048849
  1.1.1.1(vpls-id 100)      rmt   Up     Jan  3 16:46:26 2010           1

    Remote PE: 1.1.1.1, Negotiated control-word: No
    Incoming label: 800009, Outgoing label: 301296
   Local interface: vt-2/1/0.1048850, Status: Up, Encapsulation: ETHERNET

      Description: Intf - vpls H-VPLS neighbor 1.1.1.1 vpls-id 100
  Connection History:
      Jan  3 16:46:26 2010  status update timer
      Jan  3 16:46:26 2010  PE route changed
      Jan  3 16:46:26 2010  In lbl Update                     800009
      Jan  3 16:46:26 2010  Out lbl Update                    301296
      Jan  3 16:46:26 2010  In lbl Update                     800009
      Jan  3 16:46:26 2010  loc intf up             vt-2/1/0.1048850
```

Related Documentation

- Example: Configuring BGP-Based H-VPLS Using Different Mesh Groups for Each Spoke Router on page 5
- VPLS Versions Overview on page 1