

Technology Overview

Configuring Hierarchical VPLS

Release

11.4



Published: 2011-11-08

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

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Technology Overview Configuring Hierarchical VPLS

Release 11.4

Copyright © 2011, Juniper Networks, Inc.

All rights reserved.

Revision History

October 2011—R1 Junos OS 11.4

The information in this document is current as of the date listed in the revision history.

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

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

Introduction	1
VPLS Versions Overview	3
LDP-based VPLS Challenges	3
H-VPLS Implementation	4
Mesh Group Configuration Options	6
Example: Configuring BGP-Based H-VPLS Using Different Mesh Groups for Each Spoke Router	9
Example: Configuring LDP-Based H-VPLS Using a Single Mesh Group to Terminate the Layer 2 Circuits	27

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>.

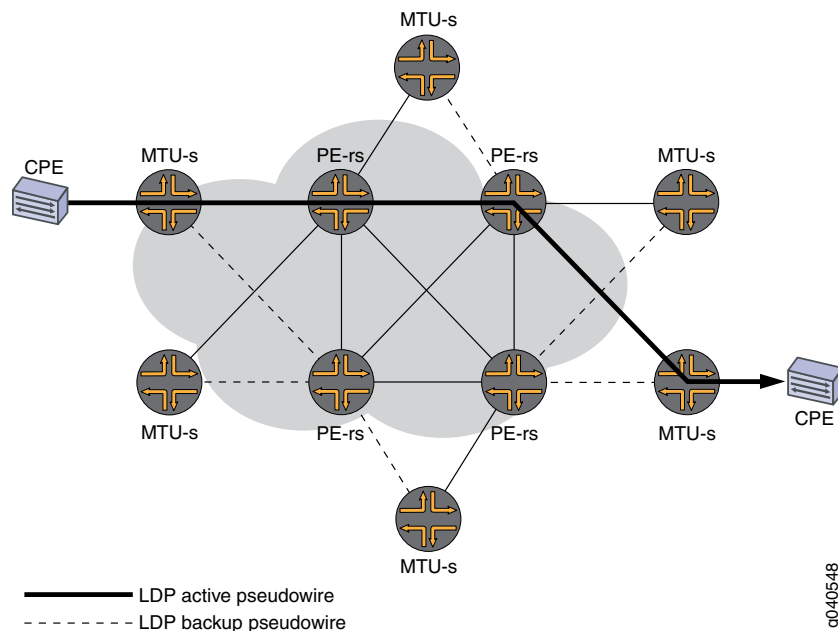
H-VPLS Implementation

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 9](#)
- [Example: Configuring LDP-Based H-VPLS Using a Single Mesh Group to Terminate the Layer 2 Circuits on page 27](#)

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

This example shows how to configure hierarchical virtual private LAN service (H-VPLS) using different mesh groups to provide H-VPLS functionality and provides steps for verifying and troubleshooting 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 27.

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 on page 9](#)
- [Overview and Topology on page 9](#)
- [Configuration on page 11](#)
- [Verification on page 24](#)

Requirements

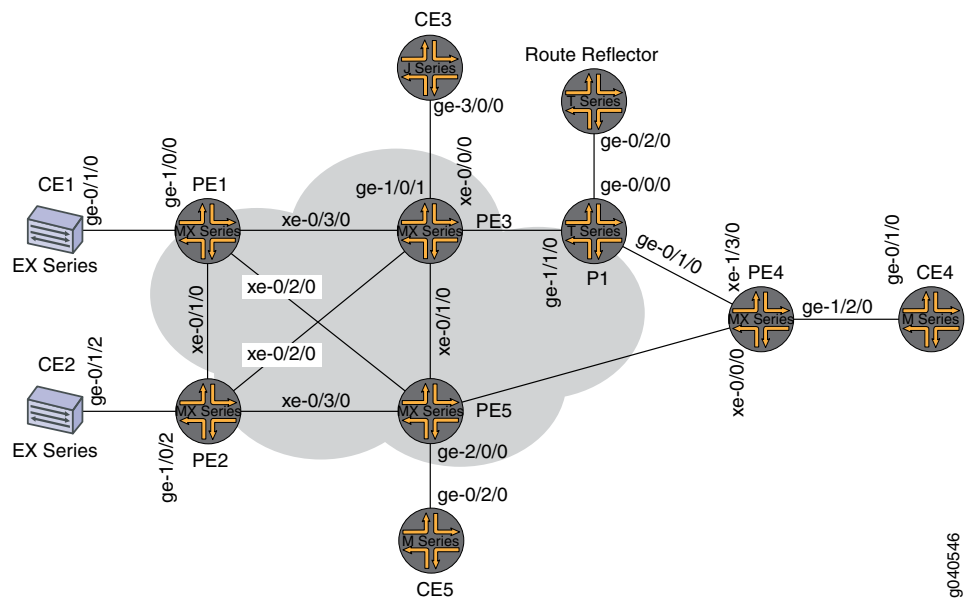
This example uses the following hardware components:

- Four MX Series 3D Universal Edge Routers for Router PE1, Router PE2, Router PE3, and Router PE4
- Two M Series Multiservice Edge Routers for Router CE4 and Router PE5
- Two EX Series Ethernet Switches for Device CE1 and Device 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 2 shows the physical topology used in this example.

Figure 2: Physical Topology of H-VPLS

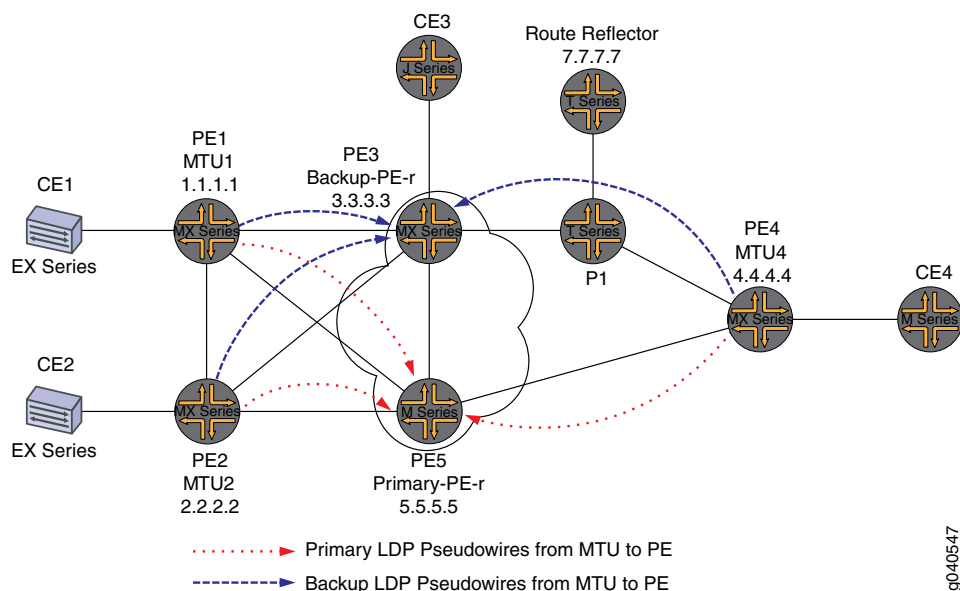


The following describes the base configuration used in this example:

- Router PE1, Router PE2, and Router PE4 are configured as MTU devices.
- Router PE3 and Router PE5 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.
- The VPLS routing instance is configured on PE-r routers with the **no-tunnel-interface** statement. This allows the MX Series routers to use a label-switched interface (LSI).
- The M320 router has a tunnel PIC installed.
- All of the routers are configured with loopback IP addresses and the autonomous system number is 65000.
- BGP is configured on the PE-r routers and the route reflector. 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 shows the logical topology used in this example.

Figure 3: Logical Topology of H-VPLS



9040547

In Figure 3 on page 11:

- Router PE1, Router PE2, and Router PE4 are configured as MTU devices. All of the MTU devices have Layer 2 circuit connections to the PE-r routers. For redundancy, a backup neighbor is configured for the Layer 2 circuit connections to the PE-r routers.
- It is not necessary to enable VPLS on the MTU devices.
- The VPLS routing instance is only configured on the PE-r routers.
- On the PE-r routers, a mesh group is created under the **H-VPLS** routing instance to terminate the Layer 2 circuit connections.
- It is not necessary to include the **l2circuit** statement in the **[edit protocols]** hierarchy on the PE-r routers. The mesh group configuration under the VPLS routing instance terminates the Layer 2 circuit pseudowires from all MTU devices in the VPLS domain.
- Each MTU device can be configured with a different virtual circuit ID or the same ID, within a single VPLS domain. The mesh groups configuration allows you to use different VPLS ID values for each mesh group.

Configuration

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

- [Configuring the Spoke PE Routers on page 12](#)
- [Configuring the Hub PE \(PE-r\) on page 13](#)
- [Verifying the H-VPLS Operation on page 18](#)

Configuring the Spoke 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-1/0/0 {
  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 PE5 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 IP address of Router PE3 as the backup neighbor, and including the **standby** statement.

```
[edit protocols]
l2circuit {
  neighbor 5.5.5.5 {
    interface ge-1/0/0.0 {
      virtual-circuit-id 100;
      backup-neighbor 3.3.3.3 { # 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-1/0/2 {
  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 PE5 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 IP address of Router PE3 as the backup neighbor, and including the **standby** statement.

```
[edit protocols]
```



```
l2circuit {
  neighbor 5.5.5.5 {
    interface ge-1/0/2.0 {
      virtual-circuit-id 200; # different VC-ID
      encapsulation-type ethernet; # default encapsulation
      backup-neighbor 3.3.3.3 {
        standby;
      }
    }
  }
}
```

5. On Router PE4, configure the Gigabit Ethernet interface connected to Router CE4. 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.

```
ge-1/2/0 {
  encapsulation ethernet-ccc;
  unit 0 {
    family ccc;
  }
}
```

6. On Router PE4, configure the Layer 2 circuit by including the **neighbor** statement and specifying the IP address of Router PE5 as the neighbor. Configure the Gigabit Ethernet logical interface by including the **virtual-circuit-id** statement and specifying **400** as the ID. Also configure a backup neighbor for the Layer 2 circuit by including the **backup-neighbor** statement, specifying the IP address of Router PE3 as the backup neighbor, and including the **standby** statement.

```
[edit protocols]
l2circuit {
  neighbor 5.5.5.5 {
    interface ge-1/2/0.0 {
      virtual-circuit-id 400;
      backup-neighbor 3.3.3.3 {
        standby;
      }
    }
  }
}
```

Configuring the Hub PE (PE-r)

Step-by-Step Procedure

1. On Router PE5 (the primary hub), configure the Gigabit Ethernet interface connected to Router CE5. Include the **encapsulation** statement and specify the **ethernet-vpls** option. Also configure the logical interface by including the **family inet** statement and specifying the IPv4 address for the interface.

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

```
    }  
  }  
  lo0 {  
    unit 0 {  
      family inet {  
        address 5.5.5.5/32;  
      }  
    }  
  }  
}
```

2. On PE-r Router PE5, 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 CE5. Configure a route distinguisher to ensure that the route advertisement is unique by including the **route-distinguisher** statement and specifying **7.7.7.77** 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:65000:2** as the value.

```
[edit]  
routing-instances {  
  H-VPLS {  
    instance-type vpls;  
    interface ge-2/0/0.0;  
    route-distinguisher 7.7.7.77;  
    vrf-target target:65000:2;  
  }  
}
```

3. On PE-r Router PE5, configure a provider tunnel that makes use of dynamic point-to-multipoint LSPs by including the **provider-tunnel** statement at the **[edit routing-instances H-VPLS]** hierarchy level. Configure a dynamic label switched path that uses resource reservation protocol (RSVP) signaling to dynamically create the LSP. To configure the LSP, include the **label-switched-path-template** statement at the **[edit routing-instances H-VPLS provider-tunnel]** hierarchy level and specify **vpls-GOLD-p2mp-template** as the name of the template to use.

The configuration of the **vpls-GOLD-p2mp-template** template is shown in the results section of this example.

```
[edit]  
routing-instances {  
  H-VPLS {  
    provider-tunnel {  
      rsvp-te {  
        label-switched-path-template {  
          vpls-GOLD-p2mp-template;  
        }  
      }  
    }  
  }  
}
```

4. On PE-r Router PE5, configure the VPLS protocol and the mesh groups for each of the spoke PE routers. It is not necessary to configure the Layer 2 circuit (L2-circuit) protocol on the hub PE. Configuring mesh groups under the VPLS instance terminates the Layer 2 circuit into the VPLS instance without the use of a logical tunnel interface.

To configure the VPLS protocol, include the **vpls** statement at the **[edit routing-instances H-VPLS protocols]** hierarchy level. Include the **site-range** statement and specify **8** as the value. Include the **no-tunnel-services** statement to enable the use of LSI interfaces. Include the **site** statement and specify **CE5** as the name of the site. Include the **interface** statement and specify the Gigabit Ethernet interface connected to CE5.

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 spoke PE router associated with each mesh group. Include the **vpls-id** statement and specify the site ID that matches the virtual circuit ID configured in the *Configuring the Spoke PE Routers* section of this example. Also include the **neighbor** statement and specify the IP address of the spoke PE router associated with each mesh group. For the mesh group for Router PE2, include the **encapsulation-type** statement and specify the **ethernet** option.

```
[edit routing-instances H-VPLS]
protocols {
  vpls {
    site-range 8;
    site CE5 {
      site-identifier 5;
      interface ge-2/0/0.0;
    }
    mesh-group pe4 {
      vpls-id 400;
      neighbor 4.4.4.4;
    }
    mesh-group pe2 {
      vpls-id 200;
      neighbor 2.2.2.2 {
        encapsulation-type ethernet;
      }
    }
    mesh-group pe1 {
      vpls-id 100;
      neighbor 1.1.1.1;
    }
  }
}
```

5. On Router PE3 (the backup hub), configure the Gigabit Ethernet interface connected to Router CE3 by including the **encapsulation** statement and specifying the **ethernet-ccc** option. Also configure the logical interface. Include the **family inet** statement and specify the IP address for the interface.

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

```

        family vpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 3.3.3.3/32;
        }
    }
}

```

6. 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. To configure the route distinguisher, include the **route-distinguisher** statement and specify **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:65000:2** as the value.

```

[edit routing-instances]
H-VPLS {
    instance-type vpls;
    interface ge-1/0/1.0;
    route-distinguisher 3.3.3.3:33;
    vrf-target target:65000:2;
}

```

7. On PE-r Router PE3, configure a provider tunnel that makes use of dynamic point-to-multipoint LSPs by including the **provider-tunnel** statement at the **[edit routing-instances H-VPLS]** hierarchy level. Configure a dynamic LSP that uses resource reservation protocol (RSVP) signaling to dynamically create the LSP. To configure the LSP, include the **label-switched-path-template** statement at the **[edit routing-instances H-VPLS provider-tunnel]** hierarchy level and specify **vpls-GOLD-p2mp-template** as the name of the template to use.

The configuration of the **vpls-GOLD-p2mp-template** template is shown in the results section of this example.

```

[edit routing-instances H-VPLS]
provider-tunnel {
    rsvp-te {
        label-switched-path-template {
            vpls-GOLD-p2mp-template;
        }
    }
}

```

8. On PE-r Router PE3, configure the VPLS protocol and the mesh groups for each of the spoke PE routers. It is not necessary to configure the Layer 2 circuit (L2-circuit) protocol on the Hub PE. Configuring mesh groups under the VPLS instance terminates the Layer 2 circuit into the VPLS instance without the use of a logical tunnel interface.

To configure the VPLS protocol, include the `vpls` statement at the **[edit routing-instances H-VPLS protocols]** hierarchy level. Include the `site-range` statement and specify **8** as the value. Include the `no-tunnel-services` statement to enable the use of LSI interfaces. Include the `site` statement and specify **mtu-pe4** as the name of the site. Include the `interface` statement and specify the Gigabit Ethernet interface connected to CE3.

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 spoke PE router associated with each mesh group. Include the `vpls-id` statement and specify the site ID that matches the virtual circuit ID configured in the *Configuring the Spoke PE Routers* section of this example. Also include the `neighbor` statement and specify the IP address of the spoke PE router associated with each mesh group.

```
[edit routing-instances H-VPLS]
protocols {
  vpls {
    site-range 8;
    no-tunnel-services;
    site mtu-pe4 {
      site-identifier 3;
      interface ge-1/0/1.0;
    }
    mesh-group pe4 {
      vpls-id 400;
      neighbor 4.4.4.4;
    }
    mesh-group pe2 {
      vpls-id 200;
      neighbor 2.2.2.2;
    }
    mesh-group pe1 {
      vpls-id 100;
      neighbor 1.1.1.1;
    }
  }
}
```

Verifying the H-VPLS Operation

Step-by-Step Procedure This section describes the show commands you can use to validate that the H-VPLS is working as expected.

1. On Router PE1, use the **show l2circuit connections** command to verify that the Layer 2 circuit to Router PE5 is **Up** and the Layer 2 circuit to Router PE3 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	XX -- unknown
SP -- Static Pseudowire	

Legend for interface status

Up -- operational

Dn -- down

Neighbor: 3.3.3.3

Interface	Type	St	Time last up	# Up trans
ge-1/0/0.0(vc 100)	rmt	ST		

Neighbor: 5.5.5.5

Interface	Type	St	Time last up	# Up trans
ge-1/0/0.0(vc 100)	rmt	Up	Jan 2 14:52:20 2010	1

Remote PE: 5.5.5.5, Negotiated control-word: No

Incoming label: 301296, Outgoing label: 800005

Local interface: ge-1/0/0.0, Status: Up, Encapsulation: ETHERNET

2. On Router PE1, 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
3.3.3.3	lo0.0	3.3.3.3:0	40
5.5.5.5	lo0.0	5.5.5.5:0	37

3. On Router PE5, 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@PE5# show vpls connections
```

Instance: H-VPLS

BGP-VPLS State <<<Local CE connected through BGP-based VPLS PE router

Local site: mtu-pe4 (3)

connection-site	Type	St	Time last up	# Up trans
-----------------	------	----	--------------	------------

```

5                               rmt  Up    Jan  2 21:27:20 2010          1
  Remote PE: 5.5.5.5, Negotiated control-word: No
  Incoming label: 262165, Outgoing label: 800258
  Local interface: lsi.1057801, Status: Up, Encapsulation: VPLS
  Description: Intf - vpls H-VPLS local site 3 remote site 5
LDP-VPLS State <<<Layer 2 circuit terminated in VPLS using mesh groups
Mesh-group connections: pe4 <<<mesh group
Neighbor                Type St    Time last up          # Up trans
4.4.4.4(vpls-id 400)    rmt  Up    Jan  2 15:47:13 2010          1
  Remote PE: 4.4.4.4, Negotiated control-word: No
  Incoming label: 262409, Outgoing label: 301088
  Local interface: lsi.1057796, Status: Up, Encapsulation: ETHERNET
  Description: Intf - vpls H-VPLS neighbor 4.4.4.4 vpls-id 400
Mesh-group connections: pe2
Neighbor                Type St    Time last up          # Up trans
2.2.2.2(vpls-id 200)    rmt  Up    Jan  2 21:04:40 2010          1
  Remote PE: 2.2.2.2, Negotiated control-word: No
  Incoming label: 262410, Outgoing label: 301488
  Local interface: lsi.1057797, Status: Up, Encapsulation: ETHERNET
  Description: Intf - vpls H-VPLS neighbor 2.2.2.2 vpls-id 200
Mesh-group connections: pe1
Neighbor                Type St    Time last up          # Up trans
1.1.1.1(vpls-id 100)    rmt  Up    Jan  2 15:47:13 2010          1
  Remote PE: 1.1.1.1, Negotiated control-word: No
  Incoming label: 262411, Outgoing label: 301328
  Local interface: lsi.1057798, Status: Up, Encapsulation: ETHERNET
  Description: Intf - vpls H-VPLS neighbor 1.1.1.1 vpls-id 100

```

4. On Router PE5, use the **show ldp neighbor** command to verify that a targeted LDP session has been created to each of the spoke PE routers (MTUs).

```
user@PE5# show ldp neighbor
```

Address	Interface	Label space ID	Hold time
1.1.1.1	lo0.0	1.1.1.1:0	41
2.2.2.2	lo0.0	2.2.2.2:0	44
4.4.4.4	lo0.0	4.4.4.4:0	32

5. On Router PE5, use the **show vpls mac-table** command to verify that MAC addresses of Router CE1, Router CE2, and Router CE3 have been learned.

```
user@PE5# show vpls mac-table
```

```
MAC flags (S -static MAC, D -dynamic MAC,
SE -Statistics enabled, NM -Non configured MAC)
```

```

Routing instance : H-VPLS
Bridging domain : __H-VPLS__, VLAN : NA
MAC              MAC              Logical
address          flags      interface
00:10:db:e9:4e:b6 D        ge-1/0/1.0    <<<Local Site MAC
00:12:1e:c6:98:3e D        lsi.1057801    <<<CE1 MAC
00:14:f6:75:78:1f D        lsi.1057801    <<<CE3 MAC
00:1f:12:32:b1:d8 D        lsi.1057801    <<<CE2 MAC

```

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

The relevant sample configuration for the spoke Router PE1 follows.

```
Router PE1 interfaces {
  xe-0/1/0 {
    unit 0 {
      family inet {
        address 10.10.2.1/30;
      }
      family mpls;
    }
  }
  xe-0/2/0 {
    unit 0 {
      family inet {
        address 10.10.3.1/30;
      }
      family mpls;
    }
  }
  xe-0/3/0 {
    unit 0 {
      family inet {
        address 10.10.1.1/30;
      }
      family mpls;
    }
  }
  ge-1/0/0 {
    encapsulation ethernet-ccc;
    unit 0 {
      family ccc;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 1.1.1.1/32;
      }
    }
  }
}
routing-options {
  static {
    route 172.0.0.0/8 next-hop 172.19.59.1;
  }
  autonomous-system 65000;
}
protocols {
  mpls {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface all;
    }
  }
}
```



```

        interface fxp0.0 {
            disable;
        }
    }
}
ldp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
l2circuit {
    neighbor 5.5.5.5 {
        interface ge-1/0/0.0 {
            virtual-circuit-id 100;
            backup-neighbor 3.3.3.3 {
                standby;
            }
        }
    }
}
}
}

```

The relevant sample configuration for Router PE3 follows.

```

Router PE3  interfaces {
                xe-0/0/0 {
                    unit 0 {
                        family inet {
                            address 10.10.20.2/30;
                        }
                        family mpls;
                    }
                }
                xe-0/1/0 {
                    unit 0 {
                        family inet {
                            address 10.10.6.1/30;
                        }
                        family mpls;
                    }
                }
                xe-0/2/0 {
                    unit 0 {
                        family inet {
                            address 10.10.5.2/30;
                        }
                        family mpls;
                    }
                }
                xe-0/3/0 {
                    unit 0 {
                        family inet {
                            address 10.10.1.2/30;
                        }
                        family mpls;
                    }
                }
            }

```

```
    }
  }
  ge-1/0/1 {
    encapsulation ethernet-vpls;
    unit 0 {
      family vpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 3.3.3.3/32;
      }
    }
  }
}
routing-options {
  static {
    route 172.0.0.0/8 next-hop 172.19.59.1;
  }
  autonomous-system 65000;
}
protocols {
  rsvp {
    interface all;
    interface fxp0.0 {
      disable;
    }
    interface xe-0/0/0.0 {
      link-protection;
    }
    interface xe-0/1/0.0 {
      link-protection;
    }
    interface xe-0/3/0.0 {
      link-protection;
    }
    interface xe-0/2/0.0 {
      link-protection;
    }
  }
}
mpls {
  label-switched-path to-RR {
    to 7.7.7.7;
  }
  label-switched-path vpls-GOLD-p2mp-template {
    template;
    optimize-timer 50;
    link-protection;
    p2mp;
  }
  label-switched-path to-PE2 {
    to 2.2.2.2;
  }
  label-switched-path to-PE3 {
    to 3.3.3.3;
  }
}
```

```
}
label-switched-path to-PE4 {
  to 4.4.4.4;
}
label-switched-path to-PE1 {
  to 1.1.1.1;
}
interface all;
interface fxp0.0 {
  disable;
}
}
bgp {
  group RR {
    type internal;
    local-address 3.3.3.3;
    family l2vpn {
      signaling;
    }
    neighbor 7.7.7.7;
  }
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}
ldp {
  interface all;
  interface fxp0.0 {
    disable;
  }
}
}
routing-instances {
  H-VPLS {
    instance-type vpls;
    interface ge-1/0/1.0;
    route-distinguisher 3.3.3.3:33;
    provider-tunnel {
      rsvp-te {
        label-switched-path-template {
          vpls-GOLD-p2mp-template;
        }
      }
    }
  }
  vrf-target target:65000:2;
  protocols {
    vpls {
      site-range 8;
      no-tunnel-services;
      site mtu-pe4 {
```

```
        site-identifier 3;
        interface ge-1/0/1.0;
    }
    mesh-group pe4 {
        vpls-id 400;
        neighbor 4.4.4.4;
    }
    mesh-group pe2 {
        vpls-id 200;
        neighbor 2.2.2.2;
    }
    mesh-group pe1 {
        vpls-id 100;
        neighbor 1.1.1.1;
    }
}
}
```

Verification

To confirm that the complete configuration is working properly, perform these tasks:

- [Verifying VPLS Connections From Router CE1 on page 24](#)
- [Verifying VPLS Connections From Router CE3 on page 24](#)

Verifying VPLS Connections From Router CE1

Purpose To verify the CE-to-CE VPLS connections from Router CE1.

Action Use the **ping** command to verify connectivity from Router CE1 to Router CE2, Router CE3, Router CE4, and Router CE5.

```
user@CE1# ping 40.40.40.2
PING 40.40.40.2 (40.40.40.2): 56 data bytes
64 bytes from 40.40.40.2: icmp_seq=0 ttl=64 time=2.513 ms
64 bytes from 40.40.40.2: icmp_seq=1 ttl=64 time=1.940 ms

user@CE1# ping 40.40.40.3
PING 40.40.40.3 (40.40.40.3): 56 data bytes
64 bytes from 40.40.40.3: icmp_seq=0 ttl=64 time=0.943 ms
64 bytes from 40.40.40.3: icmp_seq=1 ttl=64 time=0.868 ms

user@CE1# ping 40.40.40.5
PING 40.40.40.5 (40.40.40.5): 56 data bytes
64 bytes from 40.40.40.5: icmp_seq=0 ttl=64 time=1.196 ms
64 bytes from 40.40.40.5: icmp_seq=1 ttl=64 time=17.260 ms

user@CE1# ping 40.40.40.11
PING 40.40.40.11 (40.40.40.11): 56 data bytes
64 bytes from 40.40.40.11: icmp_seq=0 ttl=64 time=1.027 ms
64 bytes from 40.40.40.11: icmp_seq=1 ttl=64 time=1.013 ms
```

Verifying VPLS Connections From Router CE3

Purpose To verify the CE-to-CE VPLS connections from Router CE3.

Action Use the **ping** command to verify connectivity from Router CE3 to Router CE1, Router CE2, Router CE4, and Router CE5.

```
user@CE3> ping 40.40.40.1
PING 40.40.40.1 (40.40.40.1): 56 data bytes
64 bytes from 40.40.40.1: icmp_seq=0 ttl=64 time=1.999 ms
64 bytes from 40.40.40.1: icmp_seq=1 ttl=64 time=1.175 ms

user@CE3> ping 40.40.40.2
PING 40.40.40.2 (40.40.40.2): 56 data bytes
64 bytes from 40.40.40.2: icmp_seq=0 ttl=64 time=3.483 ms
64 bytes from 40.40.40.2: icmp_seq=1 ttl=64 time=1.170 ms

user@CE3> ping 40.40.40.5
PING 40.40.40.5 (40.40.40.5): 56 data bytes
64 bytes from 40.40.40.5: icmp_seq=0 ttl=64 time=2.813 ms
64 bytes from 40.40.40.5: icmp_seq=1 ttl=64 time=1.170 ms

user@CE3> ping 40.40.40.11
PING 40.40.40.11 (40.40.40.11): 56 data bytes
64 bytes from 40.40.40.11: icmp_seq=0 ttl=64 time=2.125 ms
64 bytes from 40.40.40.11: icmp_seq=2 ttl=64 time=124.979 ms
```

- Related Documentation**
- [Example: Configuring LDP-Based H-VPLS Using a Single Mesh Group to Terminate the Layer 2 Circuits on page 27](#)
 - [VPLS Versions Overview on page 3](#)

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 9](#).

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 on page 27](#)
- [Overview and Topology on page 27](#)
- [Configuration on page 28](#)

Requirements

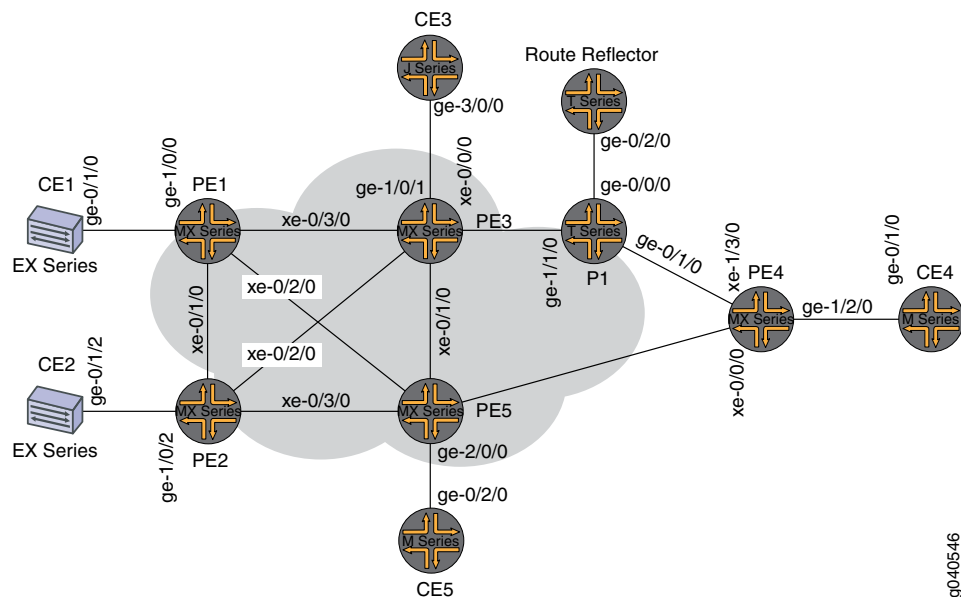
This example uses the following hardware components:

- 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 28](#) 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 28:

- 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 28](#)
- [Configuring the Hub PE Router on page 30](#)
- [Verification on page 31](#)

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;
    }
  }
}
```

- 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

- 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

- 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 9](#)
- [VPLS Versions Overview on page 3](#)