

# Network Configuration Example

## Interconnecting a Layer 2 Circuit with a Layer 2 Circuit

Release  
**12.3**



---

Published: 2012-11-14

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.

*Network Configuration Example Interconnecting a Layer 2 Circuit with a Layer 2 Circuit*

Release 12.3

Copyright © 2012, Juniper Networks, Inc.

All rights reserved.

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

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

Layer 2 Circuit Overview . . . . .	1
Applications for Interconnecting a Layer 2 Circuit with a Layer 2 Circuit . . . . .	2
Example: Interconnecting a Layer 2 Circuit with a Layer 2 Circuit . . . . .	2



## Layer 2 Circuit Overview

---

A Layer 2 circuit is a point-to-point Layer 2 connection transported using Multiprotocol Label Switching (MPLS) or other tunneling technology on the service provider's network. A Layer 2 circuit is similar to a circuit cross-connect (CCC), except that multiple virtual circuits (VCs) are transported over a single shared label-switched path (LSP) tunnel between two provider edge (PE) routers. In contrast, each CCC requires a separate dedicated LSP.

To establish a Layer 2 circuit, the Link Integrity Protocol (LIP) is used as the signaling protocol to advertise the ingress label to the remote PE routers. For this purpose, a targeted remote LDP neighbor session is established using the extended discovery mechanism described in LDP, and the session is brought up to the remote PE loopback IP address. Because LDP looks at the Layer 2 circuit configuration and initiates extended neighbor discovery for all the Layer 2 circuit neighbors (the remote PEs), no new configuration is necessary in LDP. Each Layer 2 circuit is represented by the logical interface connecting the local PE router to the local customer edge (CE) router. Note that LDP must be enabled on the lo0.0 interface for extended neighbor discovery to function correctly.

Packets are sent to remote CE routers over an egress VPN label advertised by the remote PE router, using a targeted LDP session. The VPN label is sent over an LDP LSP to the remote PE router connected to the remote CE router. Return traffic from the remote CE router destined to the local CE router is sent using an ingress VPN label advertised by the local PE router, which is also sent over the LDP LSP to the local PE router from the remote PE router.

### Related Documentation

- [Layer 3 VPN Overview](#)
- [Layer 2 VPN Overview](#)
- [Layer 2 VPN Applications](#)
- [Applications for Interconnecting a Layer 2 Circuit with a Layer 2 Circuit on page 2](#)
- [Applications for Interconnecting a Layer 2 Circuit with a Layer 3 VPN](#)
- [Example: Interconnecting a Layer 2 Circuit with a Layer 2 Circuit on page 2](#)
- [Example: Interconnecting a Layer 2 Circuit with a Layer 3 VPN](#)
- [Example: Interconnecting a Layer 2 Circuit with a Layer 2 VPN](#)

## Applications for Interconnecting a Layer 2 Circuit with a Layer 2 Circuit

---

MPLS-based Layer 2 services are growing in demand among enterprise and service providers. This creates new challenges for service providers who want to provide end-to-end value-added services. There are various reasons to stitch different Layer 2 services to one another and to Layer 3 services, for example, to expand the service offerings and to expand geographically. The Junos OS has various features to address the needs of the service provider.

Interconnecting a Layer 2 circuit with a Layer 2 circuit includes the following benefits:

- Interconnecting a Layer 2 circuit with a Layer 2 circuit enables the sharing of a service provider's core network infrastructure between Layer 2 circuit services, reducing the cost of providing those services. A Layer 2 MPLS circuit allows service providers to create a Layer 2 circuit service over an existing IP and MPLS backbone.
- Service providers do not have to invest in separate Layer 2 equipment to provide Layer 2 circuit service. A service provider can configure a provider edge router to run any Layer 2 protocol. Customers who prefer to maintain control over most of the administration of their own networks want Layer 2 circuit connections with their service provider instead of a Layer 3 VPN connection.

### Related Documentation

- [Layer 2 Circuit Overview on page 1](#)
- [Example: Interconnecting a Layer 2 Circuit with a Layer 2 Circuit on page 2](#)

## Example: Interconnecting a Layer 2 Circuit with a Layer 2 Circuit

---

This example provides a step-by-step procedure and commands for configuring and verifying a Layer 2 circuit to a Layer 2 circuit interconnection. It contains the following sections:

- [Requirements on page 2](#)
- [Overview and Topology on page 3](#)
- [Configuration on page 4](#)

### Requirements

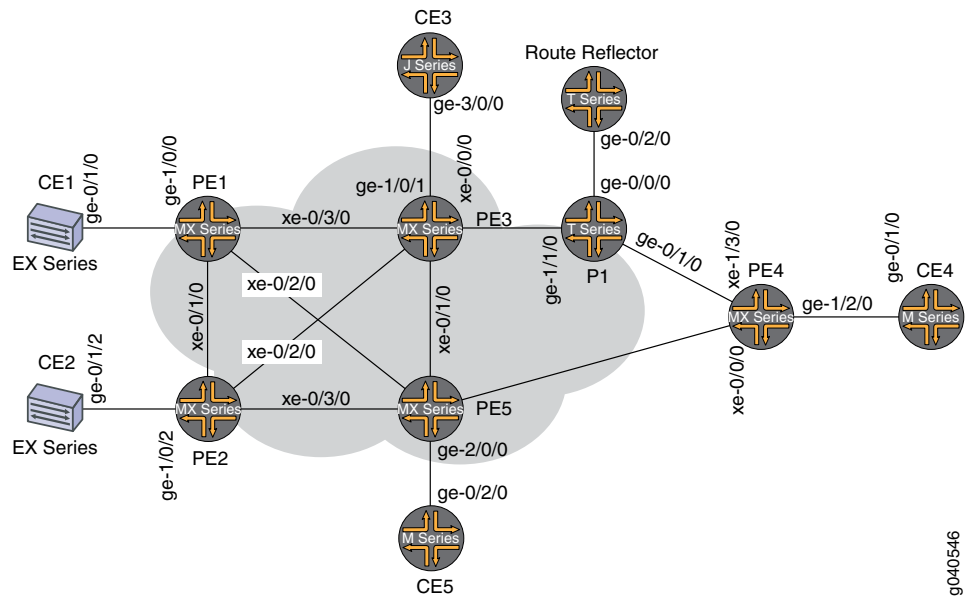
This example uses the following hardware and software components:

- Junos OS Release 9.3 or later
- 2 MX Series routers
- 2 M Series routers
- 1 T Series router
- 1 EX Series router
- 1 J Series router

## Overview and Topology

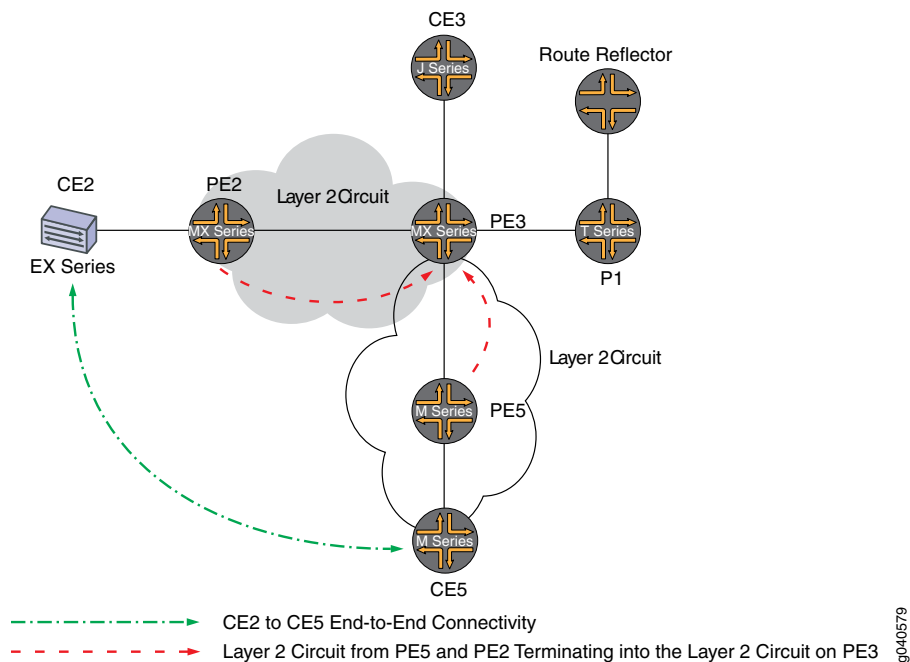
The physical topology of a Layer 2 circuit to Layer 2 circuit interconnection is shown in [Figure 1 on page 3](#)

**Figure 1: Physical Topology of a Layer 2 Circuit Terminating into a Layer 2 Circuit**



The logical topology of a Layer 2 circuit to Layer 2 circuit interconnection is shown in [Figure 2 on page 4](#)

**Figure 2: Logical Topology of a Layer 2 Circuit Terminating into a Layer 2 Circuit**



## Configuration



**NOTE:** In any configuration session, it is good practice to verify periodically that the configuration can be committed using the `commit check` command.

In this example, the router being configured is identified using the following command prompts:

- **CE2** identifies the customer edge 2 (CE2) router
- **PE1** identifies the provider edge 1 (PE1) router
- **CE3** identifies the customer edge 3 (CE3) router
- **PE3** identifies the provider edge 3 (PE3) router
- **CE5** identifies the customer edge 5 (CE5) router
- **PE5** identifies the provider edge 5 (PE5) router

This example contains the following procedures:

- [Configuring PE Router Customer-facing and Loopback Interfaces on page 5](#)
- [Configuring Core-facing Interfaces on page 6](#)
- [Configuring Protocols on page 7](#)
- [Configuring the Layer 2 Circuits on page 8](#)
- [Interconnecting the Layer 2 Circuits on page 10](#)



- [Verifying the Layer 2 Circuit to Layer 2 Circuit Interconnection on page 11](#)
- [Results on page 14](#)

### Configuring PE Router Customer-facing and Loopback Interfaces

#### Step-by-Step Procedure

To begin building the interconnection, configure the interfaces on the PE routers. If your network contains provider (P) routers, configure the interfaces on the P routers also. This example shows the configuration for Router PE1 and Router PE5.

1. On Router PE1, configure the **ge-1/0/0** interface encapsulation. To configure the interface encapsulation, include the **encapsulation** statement and specify the **ethernet-ccc** option (vlan-ccc encapsulation is also supported). Configure the **ge-1/0/0.0** logical interface family for circuit cross-connect functionality. To configure the logical interface family, include the **family** statement and specify the **ccc** option.

```
[edit interfaces]
ge-1/0/0 {
  encapsulation ethernet-ccc;
  unit 0 {
    family ccc;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 1.1.1.1/32;
    }
  }
}
```

2. On Router PE5, configure the **ge-2/0/0** interface encapsulation. To configure the interface encapsulation, include the **encapsulation** statement and specify the **ethernet-ccc** option. Configure the **ge-2/0/0.0** logical interface family for circuit cross-connect functionality. To configure the logical interface family, include the **family** statement and specify the **ccc** option

```
[edit interfaces]
ge-2/0/0 {
  encapsulation ethernet-ccc;
  unit 0 {
    family ccc;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 5.5.5.5/32;
    }
  }
}
```

3. On Router PE3, configure the logical loopback interface. The loopback interface is used to establish the targeted LDP sessions to Routers PE1 and PE5.

```
[edit interfaces]
```

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

---

### Configuring Core-facing Interfaces

#### Step-by-Step Procedure

This procedure describes how to configure the core-facing interfaces on the PE routers. This example does not include all the core-facing interfaces shown in the physical topology illustration. Enable the **mpls** and **inet** address families on the core-facing interfaces.

1. On Router PE1, configure the **xe-0/3/0** interface. Include the **family** statement and specify the **inet** address family. Include the **address** statement and specify **10.10.1.1/30** as the interface address. Include the **family** statement and specify the **mpls** address family.

```
[edit interfaces]  
xe-0/3/0 {  
  unit 0 {  
    family inet {  
      address 10.10.1.1/30;  
    }  
    family mpls;  
  }  
}
```

2. On Router PE3, configure the core-facing interfaces. Include the **family** statement and specify the **inet** address family. Include the **address** statement and specify the IPv4 addresses shown in the example as the interface addresses. Include the **family** statement and specify the **mpls** address family. In the example, the **xe-0/0/0** interface is connected to the route reflector, the **xe-0/1/0** interface is connected to Router PE5, the **xe-0/2/0** interface is connected to Router PE2, and the **xe-0/3/0** interface is connected to Router PE1.

```
[edit 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;
  }
}

```

3. On Router PE5, configure the **xe-0/1/0** interface. Include the **family** statement and specify the **inet** address family. Include the **address** statement and specify **10.10.6.2/30** as the interface address. Include the **family** statement and specify the **mpls** address family.

```

[edit interfaces]
xe-0/1/0 {
  unit 0 {
    family inet {
      address 10.10.6.2/30;
    }
    family mpls;
  }
}

```

## Configuring Protocols

### Step-by-Step Procedure

This procedure describes how to configure the protocols used in this example. If your network contains P routers, configure the protocols on the P routers also.

Configure all of the PE routers and P routers with OSPF as the IGP protocol. Enable MPLS and LDP protocols on all of the interfaces except **fxp0.0**.

1. On Router PE1, enable OSPF as the IGP. Enable the MPLS and LDP protocols on all interfaces except **fxp0.0**. LDP is used as the signaling protocol on Router PE1 for the Layer 2 circuit. The following configuration snippet shows the protocol configuration for Router PE1:

```

[edit]
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;
        }
    }
}
```

2. Configure the PE and P routers with OSPF as the IGP. Enable the MPLS and LDP protocols on all interfaces except **fxp0.0**. The following configuration snippet shows the protocol configuration for Router PE3:

```
[edit]
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;
        }
    }
}
```

---

### Configuring the Layer 2 Circuits

---

#### Step-by-Step Procedure

This procedure describes how to configure the Layer 2 circuits.



**NOTE:** In this example the **ignore-mtu-mismatch** statement is required for the circuit to come up.

1. On Router PE1, configure the Layer 2 circuit. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE3 as

---

the neighbor. Include the interface statement and specify **ge-1/0/0.0** as the logical interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **100** as the identifier. Include the **ignore-mtu-mismatch** statement to allow a Layer 2 circuit to be established even though the maximum transmission unit (MTU) configured on the local PE router does not match the MTU configured on the remote PE router.

```
[edit]
protocols {
  l2circuit {
    neighbor 3.3.3.3 {
      interface ge-1/0/0.0 {
        virtual-circuit-id 100;
        ignore-mtu-mismatch;
      }
    }
  }
}
```

2. On Router PE5, configure the Layer 2 circuit. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE3 as the neighbor. Include the interface statement and specify **ge-2/0/0.0** as the logical interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **200** as the identifier. Include the **ignore-mtu-mismatch** statement to allow a Layer 2 circuit to be established even though the MTU configured on the local PE router does not match the MTU configured on the remote PE router.

```
[edit]
protocols {
  l2circuit {
    neighbor 3.3.3.3 {
      interface ge-2/0/0.0 {
        virtual-circuit-id 200;
        ignore-mtu-mismatch;
      }
    }
  }
}
```

3. On Router PE3, configure the Layer 2 circuit to Router PE1. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE1 as the neighbor. Include the interface statement and specify **iw0.0** as the logical interworking interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **100** as the identifier. Include the **ignore-mtu-mismatch** statement to allow a Layer 2 circuit to be established even though the MTU configured on the local PE router does not match the MTU configured on the remote PE router.

On Router PE3, configure the Layer 2 circuit to Router PE5. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE5 as the neighbor. Include the interface statement and specify **iw0.1** as the logical interworking interface that is participating in the Layer 2 circuit. Include

the **virtual-circuit-id** statement and specify **200** as the identifier. Include the **ignore-mtu-mismatch** statement.

```
[edit protocols]
l2circuit {
  neighbor 1.1.1.1 {
    interface iw0.0 {
      virtual-circuit-id 100;
      ignore-mtu-mismatch;
    }
  }
  neighbor 5.5.5.5 {
    interface iw0.1 {
      virtual-circuit-id 200;
      ignore-mtu-mismatch;
    }
  }
}
```

---

### Interconnecting the Layer 2 Circuits

#### Step-by-Step Procedure

Router PE3 is the router that is *stitching* the Layer 2 circuits together using the interworking interface. The configuration of the peer unit interfaces is what makes the interconnection.

1. On Router PE3, configure the **iw0.0** interface. Include the **encapsulation** statement and specify the **ethernet-ccc** option. Include the **peer-unit** statement and specify the logical interface unit 1 as the peer tunnel interface.

On Router PE3, configure the **iw0.1** interface. Include the **encapsulation** statement and specify the **ethernet-ccc** option. Include the **peer-unit** statement and specify the logical interface unit 0 as the peer tunnel interface.

```
[edit interfaces]
iw0 {
  unit 0 {
    encapsulation ethernet-ccc;
    peer-unit 1;
  }
  unit 1 {
    encapsulation ethernet-ccc;
    peer-unit 0;
  }
}
```

2. On Router PE3, configure the Layer 2 interworking **l2iw** protocol. To configure the Layer 2 interworking protocol, include the **l2iw** statement at the **[edit protocols]** hierarchy level.

```
[edit]
protocols {
  l2iw;
}
```

3. On each router, commit the configuration.

```
user@host> commit check
```

---

```
configuration check succeeds
user@host> commit
```

### Verifying the Layer 2 Circuit to Layer 2 Circuit Interconnection

---

#### Step-by-Step Procedure

Verify that the Layer 2 circuit connection on Router PE1 is up, the LDP neighbors are correct, and the MPLS label operations are correct.

1. On Router PE1, use the **show l2circuit connections** command to verify that the Layer 2 circuit from Router PE1 to Router PE3 is **Up**.

```
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  Up    Jan 5 22:00:49 2010    1
Remote PE: 3.3.3.3, Negotiated control-word: Yes (Null)
Incoming label: 301328, Outgoing label: 314736
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 IPv4 address of Router PE3 is shown as the LDP neighbor.

```
user@PE1> show ldp neighbor
```

Address	Interface	Label space ID	Hold time
3.3.3.3	lo0.0	3.3.3.3:0	41

3. On Router PE 1, use the **show route table mpls.0** command to verify the Layer 2 circuit is using the LDP label to Router PE3 in both directions (Push and Pop). In the example below, the Layer 2 circuit is associated with LDP label **301328**.

```
user@PE1> show route table mpls.0
```

```
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
0          *[MPLS/0] 1w1d 08:25:39, metric 1
            Receive
1          *[MPLS/0] 1w1d 08:25:39, metric 1
            Receive
2          *[MPLS/0] 1w1d 08:25:39, metric 1
            Receive
300432     *[LDP/9] 3d 01:13:57, metric 1
            > to 10.10.2.2 via xe-0/1/0.0, Pop
```

```

300432(S=0)      *[LDP/9] 3d 01:13:57, metric 1
                  > to 10.10.2.2 via xe-0/1/0.0, Pop
300768           *[LDP/9] 3d 01:13:57, metric 1
                  > to 10.10.3.2 via xe-0/2/0.0, Pop
300768(S=0)      *[LDP/9] 3d 01:13:57, metric 1
                  > to 10.10.3.2 via xe-0/2/0.0, Pop
300912           *[LDP/9] 3d 01:13:57, metric 1
                  > to 10.10.3.2 via xe-0/2/0.0, Swap 299856
301264           *[LDP/9] 3d 01:13:53, metric 1
                  > to 10.10.1.2 via xe-0/3/0.0, Swap 308224
301312           *[LDP/9] 3d 01:13:56, metric 1
                  > to 10.10.1.2 via xe-0/3/0.0, Pop
301312(S=0)      *[LDP/9] 3d 01:13:56, metric 1
                  > to 10.10.1.2 via xe-0/3/0.0, Pop
301328           *[L2CKT/7] 02:33:26
                  > via ge-1/0/0.0, Pop Offset: 4
ge-1/0/0.0       *[L2CKT/7] 02:33:26, metric 2
                  > to 10.10.1.2 via xe-0/3/0.0, Push 314736 Offset: -4

```

4. On Router PE3, use the **show l2circuit connections** command to verify that the Layer 2 circuit from Router PE3 to Router PE5 is **Up**, that the Layer 2 circuit from Router PE3 to Router PE1 is **Up**, that the connections to Router PE1 and Router PE5 use the **iw0** interface, and that the status for both local **iw0** interfaces is **Up**.

```

user@PE3> 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: 1.1.1.1
  Interface                Type  St    Time last up          # Up trans
  -----
  iw0.0(vc 100)            rmt   Up    Jan  5 13:50:14 2010      1

  Remote PE: 1.1.1.1, Negotiated control-word: Yes (Null)
  Incoming label: 314736, Outgoing label: 301328
  Local interface: iw0.0, Status: Up, Encapsulation: ETHERNET
Neighbor: 5.5.5.5
  Interface                Type  St    Time last up          # Up trans
  -----
  iw0.1(vc 200)            rmt   Up    Jan  5 13:49:58 2010      1

  Remote PE: 5.5.5.5, Negotiated control-word: Yes (Null)
  Incoming label: 314752, Outgoing label: 300208
  Local interface: iw0.1, Status: Up, Encapsulation: ETHERNET

```

5. On Router PE3, use the **show ldp neighbor** command to verify that the correct IPv4 addresses are shown as the LDP neighbor.



```

user@PE3> show ldp neighbor
Address          Interface      Label space ID      Hold time
1.1.1.1          lo0.0          1.1.1.1:0           44
2.2.2.2          lo0.0          2.2.2.2:0           42
4.4.4.4          lo0.0          4.4.4.4:0           31
5.5.5.5          lo0.0          5.5.5.5:0           44

```

6. On Router PE3, use the **show route table mpls.0** command to verify that the **mpls.0** routing table is populated with the Layer 2 interworking routes. Notice that in this example, the router is swapping label **314736** received from Router PE1 on the **iw0.0** to label **301328**.

```

user@PE3> show route table mpls.0
mpls.0: 16 destinations, 18 routes (16 active, 2 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0          *[MPLS/0] 1w1d 08:28:24, metric 1
           Receive
1          *[MPLS/0] 1w1d 08:28:24, metric 1
           Receive
2          *[MPLS/0] 1w1d 08:28:24, metric 1
           Receive
308160     *[LDP/9] 3d 01:16:55, metric 1
           > to 10.10.1.1 via xe-0/3/0.0, Pop
308160(S=0) *[LDP/9] 3d 01:16:55, metric 1
           > to 10.10.1.1 via xe-0/3/0.0, Pop
308176     *[LDP/9] 3d 01:16:54, metric 1
           > to 10.10.6.2 via xe-0/1/0.0, Pop
308176(S=0) *[LDP/9] 3d 01:16:54, metric 1
           > to 10.10.6.2 via xe-0/1/0.0, Pop
308192     *[LDP/9] 00:21:40, metric 1
           > to 10.10.20.1 via xe-0/0/0.0, Swap 601649
           to 10.10.6.2 via xe-0/1/0.0, Swap 299856
308208     *[LDP/9] 3d 01:16:54, metric 1
           > to 10.10.5.1 via xe-0/2/0.0, Pop
308208(S=0) *[LDP/9] 3d 01:16:54, metric 1
           > to 10.10.5.1 via xe-0/2/0.0, Pop
308224     *[LDP/9] 3d 01:16:52, metric 1
           > to 10.10.20.1 via xe-0/0/0.0, Pop
308224(S=0) *[LDP/9] 3d 01:16:52, metric 1
           > to 10.10.20.1 via xe-0/0/0.0, Pop
314736     *[L2IW/6] 02:35:31, metric2 1
           > to 10.10.6.2 via xe-0/1/0.0, Swap 300208
           [L2CKT/7] 02:35:31
           > via iw0.0, Pop      Offset: 4
314752     *[L2IW/6] 02:35:31, metric2 1
           > to 10.10.1.1 via xe-0/3/0.0, Swap 301328
           [L2CKT/7] 02:35:47
           > via iw0.1, Pop      Offset: 4
iw0.0      *[L2CKT/7] 02:35:31, metric2 1
           > to 10.10.1.1 via xe-0/3/0.0, Push 301328 Offset: -4
iw0.1      *[L2CKT/7] 02:35:47, metric2 1
           > to 10.10.6.2 via xe-0/1/0.0, Push 300208 Offset: -4

```

7. Verify that Router CE1 can send traffic to and receive traffic from Router CE5 across the interconnection, using the **ping** command.

```

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=1 ttl=64 time=22.425 ms

```

```
64 bytes from 40.40.40.11: icmp_seq=2 ttl=64 time=1.299 ms
64 bytes from 40.40.40.11: icmp_seq=3 ttl=64 time=1.032 ms
64 bytes from 40.40.40.11: icmp_seq=4 ttl=64 time=1.029 ms
```

8. Verify that Router CE5 can send traffic to and receive traffic from Router CE1 across the interconnection, using the **ping** command.

```
user@CE5>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=1.077 ms
64 bytes from 40.40.40.1: icmp_seq=1 ttl=64 time=0.957 ms
64 bytes from 40.40.40.1: icmp_seq=2 ttl=64 time=1.057 ms 1.017 ms
```

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

The relevant sample configuration for Router PE1 follows.

```
Router PE1 [edit]
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;
      }
    }
  }
}
```

---

```

}
forwarding-options {
  hash-key {
    family inet {
      layer-3;
      layer-4;
    }
    family mpls {
      label-1;
      label-2;
    }
  }
}
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 3.3.3.3 {
      interface ge-1/0/0.0 {
        virtual-circuit-id 100;
        ignore-mtu-mismatch;
      }
    }
  }
}

```

The relevant sample configuration for Router PE3 follows.

```

Router PE3 [edit]
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-ccc;
    unit 0 {
      family ccc;
    }
  }
  iw0 {
    unit 0 {
      encapsulation ethernet-ccc;
      peer-unit 1;
    }
    unit 1 {
      encapsulation ethernet-ccc;
      peer-unit 0;
    }
  }
  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 {
    l2iw;
    mpls {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    ospf {
        area 0.0.0.0 {
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
    ldp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    l2circuit {
        neighbor 1.1.1.1 {
            interface iw0.0 {
                virtual-circuit-id 100;
                ignore-mtu-mismatch;
            }
        }
        neighbor 5.5.5.5 {
            interface iw0.1 {
                virtual-circuit-id 200;
                ignore-mtu-mismatch;
            }
        }
    }
}
```

**Related  
Documentation**

- [Layer 2 Circuit Overview on page 1](#)
- [Applications for Interconnecting a Layer 2 Circuit with a Layer 2 Circuit on page 2](#)

