Chapter 17
CCC and TCC Configuration Guidelines

This chapter includes the following sections:

CCC Configuration

This section discusses the following circuit cross-connect (CCC) configuration tasks:

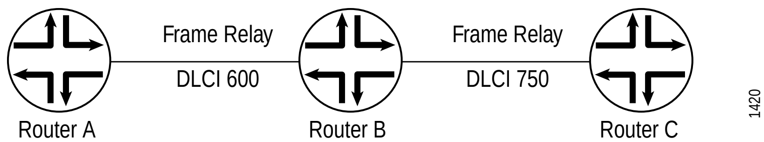
Note that you can police (control) the amount of traffic flowing over CCC circuits. For more information, refer to the *JUNOS Internet Software Configuration Guide: VPNs* .

Configure Layer 2 Switching Cross-Connects

Layer 2 switching cross-connects join logical interfaces to form what is essentially Layer 2 switching. The interfaces that you connect must be of the same type.

Figure 26 illustrates a Layer 2 switching cross-connect. In this topology, Router A and Router C have Frame Relay connections to Router B, which is a Juniper Networks router. CCC allows you to configure Router B to act as a Frame Relay (Layer 2) switch. To do this, you configure a circuit from Router A to Router C that passes through Router B, effectively configuring Router B as a Frame Relay switch with respect to these routers. This configuration allows Router B to transparently switch packets (frames) between Router A and Router C without regard to the packets' contents or the Layer 3 protocols. The only processing that Router B performs is to translate DLCI 600 to 750.

Figure 26: Layer 2 Switching Cross-Connect



If the Router A–to–Router B and Router B–to–Router C circuits were PPP, for example, the Link Control Protocol and Network Control Protocol exchanges occur between Router A and Router C. These messages are handled transparently by Router B, allowing Router A and Router C to use various PPP options (such as header or address compression and authentication) that Router B might not support. Similarly, Router A and Router C exchange keepalives, providing circuit-to-circuit connectivity status.

You can configure Layer 2 switching cross-connects on PPP, Cisco HDLC, Frame Relay, Ethernet, and ATM circuits. In a single cross-connect, only like interfaces can be connected.

To configure Layer 2 switching cross-connects, you must configure the following on the router that is acting as the switch (Router B in Figure 26):

- Define the CCC Encapsulation for Layer 2 Switching Cross-Connects on page 234
- Define the CCC Connection for Layer 2 Switching Cross-Connects on page 237
- Configure MPLS on page 237

Define the CCC Encapsulation for Layer 2 Switching Cross-Connects

To configure Layer 2 switching cross-connects, configure the CCC encapsulation on the router that is acting as the switch (Router B in Figure 26).

> **Note**: You cannot configure families on CCC interfaces; that is, you cannot include the family statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level.

PPP and Cisco HDL C

For PPP and Cisco HDLC circuits, specify the encapsulation in the encapsulation statement. This statement configures the entire physical device. For these circuits to work, you must configure a logical interface unit 0.

```
[edit]
interfaces {
   type-fpc/pic/port {
      encapsulation encapsulation-type;
      unit 0;
   }
}
```

Ethernet

For Ethernet circuits, specify ethernet-ccc in the encapsulation statement. This statement configures the entire physical device. For these circuits to work, you must configure a logical interface—unit 0.

Ethernet interfaces with standard Tag Protocol ID (TPID) tagging can use Ethernet CCC encapsulation. On M-series routers, one-port Gigabit Ethernet, two-port Gigabit Ethernet, four-port Gigabit Ethernet, and four-port Fast Ethernet PICs can use Ethernet CCC encapsulation. On T-series platforms, one-port Gigabit Ethernet and two-port Gigabit Ethernet PICs installed in FPC2 can use Ethernet CCC encapsulation. When you use this encapsulation type, you can configure the family ccc only.

```
[edit]
interfaces {
   fe-fpc/pic/port {
      encapsulation ethernet-ccc;
      unit 0;
   }
}
```

ATM

For ATM circuits, specify the encapsulation when configuring the virtual circuit (VC). Configure each VC as a circuit or a regular logical interface.

```
[edit]
interfaces {
   at-fpc/pic/port {
      atm-options {
         vpi vpi-identifier maximum-vcs maximum-vcs;
      }
      unit logical-unit-number {
         point-to-point;       # Default interface type
         encapsulation encapsulation-type;
         vci vpi-identifier.vci-identifier;
      }
   }
}
```

Frame Relay

For Frame Relay circuits, specify the encapsulation when configuring the DLCI. Configure each DLCI as a circuit or a regular logical interface. The DLCI for regular interfaces must be in the range 1 through 511. For CCC interfaces, it must be in the range 512 through 1,022.

```
[edit]
interfaces {
    type-fpc/pic/port {
        unit logical-unit-number {
            point-to-point;        # Default interface type
            encapsulation encapsulation-type;
            dlci dlci-identifier;
        }
    }
}
```

Ethernet VLAN

An Ethernet VLAN circuit can be configured using either the vlan-ccc or extended-vlan-ccc encapsulation. For extended-vlan-ccc, you cannot configure the inet family. Only the CCC family is allowed. The vlan-ccc encapsulation supports both the inet and CCC families. Ethernet interfaces in VLAN mode can have multiple logical interfaces.

For encapsulation type vlan-ccc, VLAN IDs from 512 through 1023 are reserved for CCC VLANs, allowing you to make up to 512 VLANs for the CCC connection. For the extended-vlan-ccc encapsulation type, VLAN IDs from 0 through 1023 are valid, allowing you to make up to 1,024 VLANs for the CCC connection.

Note

Some venders use the proprietary Tag Protocol IDs (TPIDs) 0x9100 and 0x9901 to encapsulate a VMAN-tagged packet into a VLAN-CCC tunnel to interconnect a geographically separated metro Ethernet network. By configuring the extended-vlan-ccc encapsulation type, a Juniper Networks router can accept all three TPIDs (0x8100, 0x9100, and 0x9901).

Configure an Ethernet VLAN circuit with the vlan-ccc encapsulation as follows:

```
[edit]
interfaces {
    type-fpc/pic/port {
        vlan-tagging;
        encapsulation vlan-ccc;
        unit logical-unit-number {
            encapsulation vlan-ccc;
            family ccc;
            vlan-id vlan-id;
        }
    }
}
```

Configure an Ethernet VLAN circuit with the extended-vlan-ccc encapsulation as follows:

```
[edit]
interfaces {
    type-fpc/pic/port {
        vlan-tagging;
        encapsulation extended-vlan-ccc;
        unit logical-unit-number {
            vlan-id vlan-id;
            family ccc;
        }
    }
}
```

Whether you configure the encapsulation as vlan-ccc or extended-vlan-ccc, you must enable VLAN tagging by including the vlan-tagging statement.

Define the CCC Connection for Layer 2 Switching Cross-Connects

To configure Layer 2 switching cross-connects, define the connection between the two circuits. You configure this on the router that is acting as the switch (Router B in Figure 26). The connection joins the interface that comes from the circuit's source to the interface that leads to the circuit's destination. When you specify the interface names, include the logical portion of the name, which corresponds to the logical unit number. The cross-connect is bidirectional, so packets received on the first interface are transmitted out the second interface, and those received on the second interface are transmitted out the first.

```
[edit]
protocols {
    connections {
        interface-switch connection-name {
            interface interface-name.unit-number;
            interface interface-name.unit-number;
        }
    }
}
```

Configure MPLS

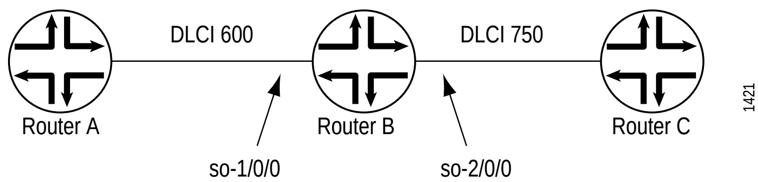
For Layer 2 switching cross-connects to work, you must configure MPLS. The following is a minimal MPLS configuration:

```
[edit]
protocols {
    mpls {
        interface (interface-name | all);
    }
}
```

Example: Configure Layer 2 Switching Cross-Connects

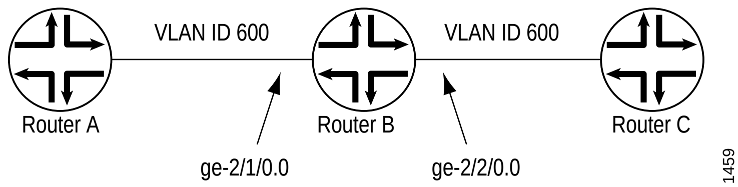
Configure a full-duplex Layer 2 switching cross-connect between Router A and Router C, using a Juniper Networks router, Router B, as the virtual switch. See the topology in Figure 27 and Figure 28.

Figure 27: Topology of a Frame Relay Layer 2 Switching Cross-Connect




```
[edit]
interfaces {
    so-1/0/0 {
        encapsulation frame-relay-ccc;
        unit 1 {
            point-to-point;
            encapsulation frame-relay-ccc;
            dlci 600;
        }
    }
    so-2/0/0 {
        encapsulation frame-relay-ccc;
        unit 2 {
            point-to-point;
            encapsulation frame-relay-ccc;
            dlci 750;
        }
    }
}
protocols {
    connections {
        interface-switch router-a-router-c {
            interface so-1/0/0.1;
            interface so-2/0/0.2;
        }
    }
    mpls {
        interface all;
    }
```

Figure 28: Sample Topology of a VLAN Layer 2 Switching Cross-Connect

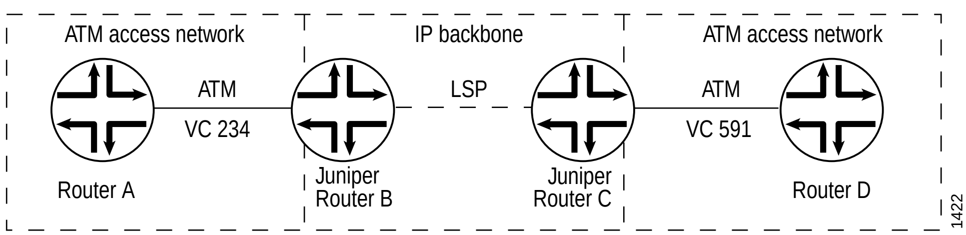



```
[edit]
interfaces {
   ge-2/1/0 {
      vlan-tagging;
      encapsulation vlan-ccc;
      unit 0 {
         encapsulation vlan-ccc;
         vlan-id 600;
      }
   }
   ge-2/2/0 {
      vlan-tagging;
      encapsulation vlan-ccc;
      unit 0 {
         encapsulation vlan-ccc;
         vlan-id 600;
      }
      unit 1 {
         family inet {
            vlan-id 1;
            address 10.9.200.1/24;
      }
}
protocols {
   mpls {
      interface all;
   }
   connections {
      interface-switch layer2-sw {
         interface ge-2/1/0.0;
         interface ge-2/2/0.0;
      }
   }
}
```

Configure MPLS LSP Tunnel Cross-Connects

MPLS tunnel cross-connects between interfaces and LSPs allow you to connect two distant interface circuits of the same type by creating MPLS tunnels that use LSPs as the conduit. The topology in Figure 29 illustrates an MPLS LSP tunnel cross-connect. In this topology, two separate networks, in this case ATM access networks, are connected through an IP backbone. CCC allows you to establish an LSP tunnel between the two domains. With LSP tunneling, you tunnel the ATM traffic from one network across a SONET backbone to the second network using an MPLS LSP.

Figure 29: MPLS LSP Tunnel Cross-Connect



When traffic from Router A (VC 234) reaches Router B, it is encapsulated and placed into an LSP, which is sent through the backbone to Router C. At Router C, the label is removed and the packets are placed onto the ATM PVC (VC 591) and sent to Router D. Similarly, traffic from Router D (VC 591) is sent over an LSP to Router B, then placed on VC 234 to Router A.

You can configure LSP tunnel cross-connects on PPP, Cisco HDLC, Frame Relay, and ATM circuits. In a single cross-connect, only like interfaces can be connected.

To configure LSP tunnel cross-connects, you must configure the following on the interdomain router (Router B in Figure 31):

Define the CCC Encapsulation for LSP Tunnel Cross-Connects on page 241

Define the CCC Connection for LSP Tunnel Cross-Connects on page 242

When you use MPLS tunnel cross-connects, if you use the default MTU size, IS-IS does not form adjacencies across the tunnel. For the tunnel cross-connects to work, the MTU size on the edge routers (Routers A and D in Figure 29) must be smaller than the LSP's MTU. Use the following calculation to determine the maximum IS-IS MTU size:

IS-IS MTU £ MPLS MTU – 4 bytes – link-layer overhead

The link-layer overheads varies, depending on the encapsulation:

ATM—8 bytes

Frame Relay—2 bytes

HDLC—4 bytes

PPP—4 bytes

VLAN—4 bytes

We recommend that you simply set the MTU to 1497 bytes, which is small enough so that IS-IS works properly.

To modify the MTU, include the mtu statement when configuring the logical interface family, at the [edit interfaces *interface-name* unit *logical-unit-number* encapsulation *family*] hierarchy level. For more information about setting the MTU, see the *JUNOS Internet Software Configuration Guide: Interfaces, Class of Service, Firewalls*.

Define the CCC Encapsulation for LSP Tunnel Cross-Connects

To configure LSP tunnel cross-connects, you must configure the CCC encapsulation on the ingress and egress routers (Router B and Router C, respectively, in Figure 29).

> **Note**
> You cannot configure families on CCC interfaces; that is, you cannot include the family statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level.

For PPP or Cisco HDLC circuits, specify the encapsulation in the encapsulation statement. This statement configures the entire physical device. For these circuits to work, you must configure a logical interface unit 0.

```
[edit]
interfaces {
  type-fpc/pic/port {
    encapsulation (ppp-ccc | cisco-hdlc-ccc);
    unit 0;
  }
}
```

For ATM circuits, specify the encapsulation when configuring the VC. For each VC, you configure whether it is a circuit or a regular logical interface.

```
[edit]
interfaces {
  at-fpc/pic/port {
    atm-options {
      vpi vpi-identifier maximum-vcs maximum-vcs;
    }
    unit logical-unit-number {
      point-to-point;        # Default interface type
      encapsulation atm-ccc-vc-mux;
      vci vpi-identifier.vci-identifier;
    }
  }
}
```

For Frame Relay circuits, specify the encapsulation when configuring the DLCI. For each DLCI, you configure whether it is a circuit or a regular logical interface. The DLCI for regular interfaces must be in the range 1 through 511. For CCC interfaces, it must be in the range 512 through 1022.

```
[edit]
interfaces {
    type-fpc/pic/port {
        encapsulation frame-relay-ccc;
        unit logical-unit-number {
            point-to-point; # default interface type
            encapsulation frame-relay-ccc;
            dlci dlci-identifier;
        }
    }
}
```

For more information about the encapsulation statement, see the *JUNOS Internet Softw are Configur ation Guide: Netw ork Interf aces and Class of Service* .

Define the CCC Connection for LSP Tunnel Cross-Connects

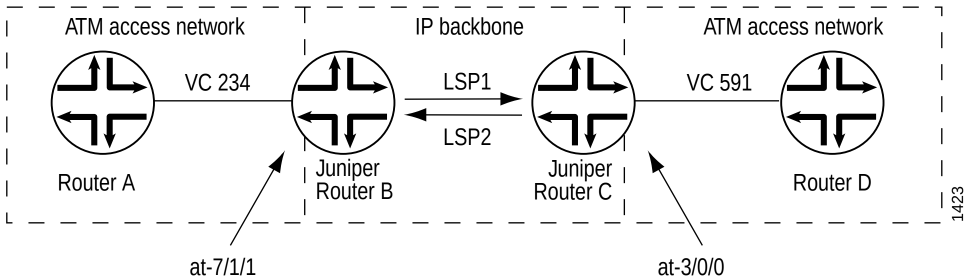
To configure LSP tunnel cross-connects, define the connection between the two circuits on the ingress and egress routers (Router B and Router C, respectively, in Figure 29). The connection joins the interface or LSP that comes from the circuit’s source to the interface or LSP that leads to the circuit’s destination. When you specify the interface name, include the logical portion of the name, which corresponds to the logical unit number. For the cross-connect to be bidirectional, you must configure cross-connects on two routers.

```
[edit]
protocols {
    connections {
        remote-interface-switch connection-name {
            interface interface-name.unit-number;
            transmit-lsp label-switched-path;
            receive-lsp label-switched-path;
        }
    }
}
```

Example: Configure LSP Tunnel Cross-Connects

Configure a full-duplex MPLS LSP tunnel cross-connect from Router A to Router D, passing through Router B and Router C. See the topology in Figure 30.

Figure 30: Example Topology of MPLS LSP Tunnel Cross-Connect



On Router B:

```
[edit]
interfaces {
   at-7/1/1 {
      atm-options {
         vpi 1 maximum-vcs 600;
      }
      unit 1 {
         point-to-point; # default interface type
         encapsulation atm-ccc-vc-mux;
         vci 1.234;
      }
   }
}
protocols {
   connections {
      remote-interface-switch router-b-to-router-c {
         interface at-7/1/1.1;
         transmit-lsp lsp1;
         receive-lsp lsp2;
      }
   }
}
```

On Router C:

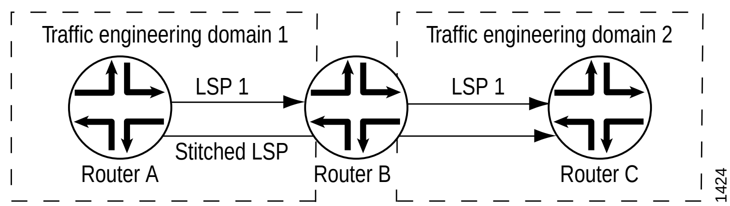
```
[edit]
interfaces {
   at-3/0/0 {
      atm-options {
         vpi 2 maximum-vcs 600;
      }
      unit 2 {
         point-to-point; # default interface type
         encapsulation atm-ccc-vc-mux;
         vci 2.591;
      }
   }
}
```

```
protocols {
    connections {
        remote-interface-switch router-b-to-router-c {
            interface at-3/0/0.1;
            transmit-lsp lsp2;
            receive-lsp lsp1;
        }
    }
}
```

Configure LSP Stitching Cross-Connects

LSP stitching cross-connects "stitch" together LSPs to join two LSPs. For example, they stitch together LSPs that fall in two different TED areas. The topology in Figure 31 illustrates an LSP stitching cross-connect. In this topology, the network is divided into two traffic engineering domains. CCC allows you to establish an LSP between the two domains by stitching together LSPs from the two domains. For LSP stitching to work, the LSPs must be dynamic LSPs, not static.

Figure 31: LSP Stitching Cross-Connect



Without LSP stitching, a packet travelling from Router A to Router C is encapsulated on Router A (the ingress router for the first LSP), decapsulated on Router B (the egress router), and then re-encapsulated on Router B (the ingress router for the second LSP). With LSP stitching, you connect LSP1 and LSP2 into a single, stitched LSP, which means that the packet is encapsulated once (on Router A) and decapsulated once (on Router C).

You can use LSP stitching to create a seamless LSP for LSPs carrying any kind of traffic.

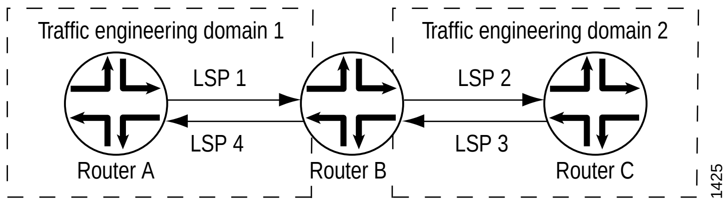
To configure LSP stitching cross-connects, configure the two LSPs that you are stitching together on the two ingress routers. Then, on the interdomain router (Router B in Figure 31), you define the connection between the two LSPs. The connection joins the LSP that comes from the connection's source to the LSP that leads to the connection's destination.

```
[edit]
protocols {
    connections {
        lsp-switch connection-name {
            transmit-lsp label-switched-path;
            receive-lsp label-switched-path;
        }
    }
}
```

Example: Configure LSP Stitching Cross-Connects

Configure a full-duplex LSP stitching cross-connect between Router A and Router C. To do this, you configure Router B, which is the interdomain router. See the topology in Figure 32.

Figure 32: Example Topology of LSP Stitching Cross-Connect




```
[edit]
protocols {
    connections {
        lsp-switch router-a-to-router-c {
            transmit-lsp lsp2;
            receive-lsp lsp1;
        }
    }
    connections {
        lsp-switch router-c-to-router-a {
            receive-lsp lsp3;
            transmit-lsp lsp4;
        }
    }
}
```

TCC Configuration

To configure TCC, you must perform the following tasks on the router that is acting as the switch:

Define the Encapsulation for the Layer 2 Switching TCCs on page 246

Define the Connection for the Layer 2 Switching TCC on page 249

Configure MPLS on page 250

Define the Encapsulation for the Layer 2 Switching TCCs

To begin configuring a Layer 2 switching TCC, configure the TCC encapsulation on the desired interfaces of the router that is acting as the switch.

> **Note:** You cannot configure standard protocol families on TCC or CCC interfaces. Only the CCC family is allowed on CCC interfaces, and only the TCC family is allowed on TCC interfaces.

You can configure the following types of circuits and encapsulations:

PPP and Cisco HDLC Circuits on page 246

ATM Circuits on page 247

Frame Relay Circuits on page 247

Ethernet Circuits on page 247

Ethernet Extended VLAN Circuits on page 248

ARP Configuration for Ethernet TCC and Ethernet Extended VLAN TCC on page 249

Note that for Ethernet circuits and Ethernet extended VLAN circuits, you also need to configure the Address Resolution Protocol (ARP). See “ARP Configuration for Ethernet TCC and Ethernet Extended VLAN TCC” on page 249.

PPP and Cisco HDLC Circuits

For PPP and Cisco HDLC circuits, specify the encapsulation in the encapsulation statement at the [edit interfaces] hierarchy level. This statement configures the entire physical device. For these circuits to work, you must configure the logical interface unit 0.

Configure PPP and Cisco HDLC circuits as follows:

```
[edit]
interfaces {
  type-fpc/pic/port {
    encapsulation (ppp-tcc | cisco-hdlc-tcc);
    unit 0;
  }
}
```

ATM Circuits

Specify the encapsulation type for ATM circuits when configuring the virtual circuit (VC). Specify whether each VC is a circuit or a regular logical interface.

Configure ATM circuits for TCC as follows:

```
[edit]
interfaces {
    at-fpc/pic/port {
        atm-options {
            vpi vpi-identifier maximum-vcs maximum-vcs;
        }
        unit logical-unit-number {
            point-to-point;
            encapsulation (atm-tcc-vc-mux | atm-tcc-snap);
            vci vpi-identifier.vci-identifier;
        }
    }
}
```

Frame Relay Circuits

Specify the encapsulation type for Frame Relay circuits when configuring the data-link connection identifier (DLCI). You configure each DLCI as a circuit or a regular logical interface. The DLCI for regular interfaces must be in the range 1 through 511. For TCC and CCC interfaces, it must be in the range 512 through 1022.

Configure Frame Relay circuits for TCC as follows:

```
[edit]
interfaces {
    encapsulation frame-relay-tcc;
    type-fpc/pic/port {
        unit logical-unit-number {
            point-to-point;
            encapsulation frame-relay-tcc;
            dlci dlci-identifier;
        }
    }
}
```

Ethernet Circuits

Specify the encapsulation type for Ethernet TCC circuits in the encapsulation statement. This statement configures the entire physical device. You must also specify a proxy address and a remote address statically at the [edit interfaces *interface-name* unit *unit-number* family tcc] hierarchy level.

The difference between the remote address and the proxy address is that the former is associated with the TCC switching router's Ethernet neighbor and the latter is associated with the TCC router's other neighbor connected by the unlike link. The remote option allows you to configure either an IP address or a Message Authentication Code (MAC) address for the Ethernet neighbor, while the proxy statement requires the IP address for the non-Ethernet neighbor.

One-port Gigabit Ethernet, two-port Gigabit Ethernet, four-port Gigabit Ethernet, and four-port Fast Ethernet PICs can use Ethernet TCC encapsulation.

Configure Ethernet circuits for TCC as follows:

```
[edit]
interfaces
   EthernetType-fpc/pic/port {
      encapsulation ethernet-tcc;
      unit 0 {
         family tcc {
            proxy {
               inet-address address;
            }
            remote {
               inet-address address;
               mac-address mac-address;
            }
         }
      }
   }
}
```

Ethernet Extended VLAN Circuits

Specify the encapsulation type for Ethernet extended VLAN circuits in the encapsulation statement. This statement configures the entire physical device. You must also enable VLAN tagging. Ethernet interfaces in VLAN mode can have multiple logical interfaces. For encapsulation type extended-vlan-tcc, all VLAN IDs from 0 through 4094 are valid, up to a maximum of 1024 VLANs. As with Ethernet circuits, you must also specify a proxy address and a remote address at the [edit interfaces *interface-name* unit *unit-number* family tcc] hierarchy level.

Configure Ethernet extended VLAN circuits for TCC as follows:

```
[edit]
interfaces {
   EthernetType-fpc/pic/port {
      vlan-tagging;
      encapsulation extended-vlan-tcc;
         unit 0 {
            vlan-id 600;
            family tcc;
               proxy {
                  inet-address address;
               }
               remote {
                  inet-address address;
                  mac-address mac-address;
               }
            }
         }
      }
   }
}
```

ARP Configuration for Ethernet TCC and Ethernet Extended VLAN TCC

All Ethernet TCC and Ethernet Extended VLAN TCC encapsulations require that you also configure the Address Resolution Protocol (ARP). Since TCC simply removes one Layer 2 header and adds another, the default form of dynamic ARP is not supported. To retain the functionality of ARP for Ethernet networks, you must configure static ARP.

You configure the arp statement at the [edit interfaces *interface-number* unit *unit-number* family inet address *ip-address*] hierarchy level. Since you already specified remote and proxy addresses on the router performing TCC switching, you must apply the static ARP statement to the Ethernet-type interfaces of the routers that connect to the TCC-switched router. The arp statement must contain the IP address and the MAC address of the remotely connected neighbor using the unlike Layer 2 protocol on the far side of the TCC switching router.

Configure static ARP as follows:

```
[edit]
interfaces
   EthernetType-fpc/pic/port {
      unit 0 {
         family inet {
            address ip-address {
               arp ip-address mac mac-address;
            }
         }
      }
   }
}
```

Define the Connection for the Layer 2 Switching TCC

You must define the connection between the two circuits of the Layer 2 switching TCC. Configure this on the router that is acting as the switch. The connection joins the interface coming from the circuit's source to the interface that leads to the circuit's destination. When you specify the interface names, include the logical portion of the name, which corresponds to the logical unit number. The cross-connect is bidirectional, so packets received on the first interface are transmitted from the second interface, and those received on the second interface are transmitted from the first.

Configure the following connection for a local interface switch:

```
[edit]
protocols {
   connections {
      interface-switch connection-name {
         interface interface-name.unit-number;
         interface interface-name.unit-number;
      }
      lsp-switch connection-name {
         transmit-lsp lsp-number;
         receive-lsp lsp-number;
      }
   }
}
```

Configure the following connection for a remote interface switch:

```
[edit]
protocols {
    connections {
        remote-interface-switch connection-name {
            interface interface-name.unit-number;
            interface interface-name.unit-number;
            transmit-lsp lsp-number;
            receive-lsp lsp-number;
        }
    }
}
```

Configure MPLS

For a Layer 2 switching cross-connect to function, you need to configure MPLS.

The following is a minimal MPLS configuration:

```
[edit]
interfaces {
    interface-name {
        unit logical-unit-number {
    }
}
protocols {
    mpls {
        interface (interface-name | all);
    }
}
```

CCC and TCC Graceful Restart Configuration

To enable CCC and TCC graceful restart, include the graceful-restart statement at the [edit routing-options] hierarchy level:

```
[edit]
routing-options {
    graceful-restart;
}
```

The graceful-restart statement enables graceful restart for all protocols supporting this feature on the router. For more information about graceful restart, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

CCC and TCC graceful restart depend upon RSVP graceful restart. If you disable RSVP graceful restart, CCC and TCC graceful restart will not work. For more information about RSVP graceful restart, see “RSVP Graceful Restart” on page 157 and “Configure RSVP Graceful Restart” on page 165.