Chapter 8
Configure Circuit Cross-Connect

Circuit cross-connect (CCC) allows you to configure transparent connections between two circuits, where a circuit can be a Frame Relay DLCI, an ATM VC, a PPP interface, a Cisco HDLC interface, or an MPLS label-switched path (LSP). Using CCC, packets from the source circuit are delivered to the destination circuit with, at most, the Layer 2 address being changed. No other processing—such as header checksums, TTL decrementing, or protocol processing—is done.

CCC circuits fall into two categories, logical interfaces, which include DLCIs, VCs, and PPP and Cisco HDLC interfaces; and LSPs. The two circuit categories provide three types of cross-connect:

Layer 2 switching—Cross-connects between logical interfaces provide what is essentially Layer 2 switching. The interfaces that you connect must be of the same type.

MPLS tunneling—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.

LSP stitching—Cross-connects between LSPs provide a way to “stitch” together two label-switched paths, including paths that fall in two different TED areas.

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.

For all CCC connections that connect interfaces, the interfaces must be of the same type; that is, ATM to ATM, Frame Relay to Frame Relay, PPP to PPP, or Cisco HDLC to Cisco HDLC.

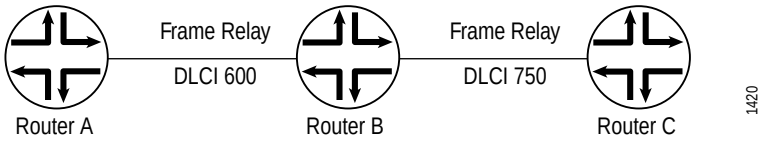
This chapter discusses the Layer 2 switching cross-connect configuration tasks. For information on MPLS tunneling and LSP stitching, see the *JUNOS Internet Software Configuration Guide: MPLS Applications*.

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 7 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 7: 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, 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 7):

Define the CCC Encapsulation for Layer 2 Switching Cross-Connects on page 102

Define the CCC Connection for Layer 2 Switching Cross-Connects on page 104

Configure MPLS on page 104

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 7).

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 {
    encapsulation frame-relay-ccc;
    type-fpc/pic/port {
        unit logical-unit-number {
            point-to-point;          # Default interface type
            encapsulation frame-relay-ccc;
            dlci dlci-identifier;
        }
    }
}
```

For Ethernet VLAN circuits, specify the encapsulation in the encapsulation statement. This statement configures the entire physical device. You must enable VLAN tagging. Ethernet interfaces in VLAN mode can have multiple logical interfaces, but in CCC mode VLAN IDs from 0 through 511 are reserved for normal VLANs and VLAN IDs from 512 through 1023 are reserved for CCC VLANs.

```
[edit]
interfaces ge-2/1/0 {
    vlan-tagging;
    encapsulation vlan-ccc;
    unit 0 {
        encapsulation vlan-ccc;
        vlan-id 600;
    }
}
```

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 7). 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

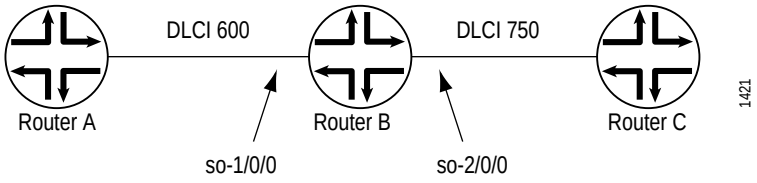
For Layer 2 switching cross-connects to work, you must configure MPLS. The following is a minimal MPLS configuration:

```
[edit]
interfaces {
    interface-name {
        unit logical-unit-number {
        }
    }
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 8.

Figure 8: Example Topology of 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;
    }
```