Chapter 17

CCC and TCC Configuration Guidelines

This chapter includes the following sections:

Configuring CCC

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 VPNs Configuration Guide*.

It is also possible to use the ping command to check the integrity of CCC LSPs. See “Pinging a CCC LSP” on page 174 for more information.

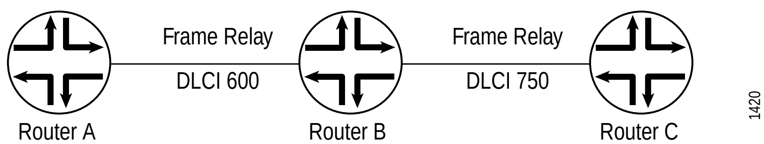
Configuring Layer 2 Switching Cross-Connects

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 27 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 configure Router B to act as a Frame Relay switch, 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 27: 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 27):

Defining the Encapsulation for Layer 2 Switching Cross-Connects on page 361

Defining the CCC Connection for Layer 2 Switching Cross-Connects on page 366

Configuring MPLS on page 366

Defining the 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 27).

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.

You can configure the following encapsulations for the Layer 2 switching cross connect:

- ATM Encapsulation for Layer 2 Switching Cross-Connects on page 361
- Ethernet Encapsulation for Layer 2 Switching Cross-Connects on page 362
- Ethernet VLAN Encapsulation for Layer 2 Switching Cross-Connects on page 362
- Aggregated Ethernet Encapsulation for Layer 2 Switching Cross-Connects on page 364
- Frame Relay Encapsulation for Layer 2 Switching Cross-Connects on page 365
- PPP and Cisco HDLC Encapsulation for Layer 2 Switching Cross-Connects on page 365

ATM Encapsulation for Layer 2 Switching Cross-Connects

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

```
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;
    }
}
```

You can include these statements at the following hierarchy levels:

[edit interfaces]

[edit logical-routers *logical-router-name* interfaces]

Ethernet Encapsulation for Layer 2 Switching Cross-Connects

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

Ethernet interfaces with standard Tag Protocol Identifier (TPID) tagging can use Ethernet CCC encapsulation. On M-series routing platforms, except the M320, one-port Gigabit Ethernet, two-port Gigabit Ethernet, four-port Gigabit Ethernet, and four-port Fast Ethernet Physical Interface Cards (PICs) can use Ethernet CCC encapsulation. On T-series platforms and M320 routers, 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 ccc family only.

```
fe-fpc/pic/port {
    encapsulation ethernet-ccc;
    unit 0;
}
```

You can include these statements at the following hierarchy levels:

[edit interfaces]

[edit logical-routers *logical-router-name* interfaces]

Ethernet VLAN Encapsulation for Layer 2 Switching Cross-Connects

An Ethernet virtual LAN (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 4094 are reserved for CCC VLANs. For the extended-vlan-ccc encapsulation type, all VLAN IDs 1 and higher are valid. VLAN ID 0 is reserved for tagging the priority of frames.

NOTE: Some vendors use the proprietary TPIDs 0x9100 and 0x9901 to encapsulate a VLAN-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:

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

You can configure these statements at the following hierarchy levels:

[edit logical-routers *logical-router-name* interfaces]

[edit interfaces]

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

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

You can configure these statements at the following hierarchy levels:

[edit logical-routers *logical-router-name* interfaces]

[edit interfaces]

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

Aggregated Ethernet Encapsulation for Layer 2 Switching Cross-Connects

You can configure aggregated Ethernet interfaces for CCC connections and for Layer 2 virtual private networks (VPNs).

Aggregated Ethernet interfaces configured with VLAN tagging can be configured with multiple logical interfaces. The only encapsulation available for aggregated Ethernet logical interfaces is vlan-ccc. When you configure the vlan-id statement, you are limited to VLAN IDs 512 through 4094.

Aggregated Ethernet interfaces configured without VLAN tagging can be configured only with the ethernet-ccc encapsulation. All untagged Ethernet packets received are forwarded based on the CCC parameters.

To configure aggregated Ethernet interfaces for CCC connections, include the ae0 statement at the [edit interfaces] hierarchy level:

```
[edit interfaces]
ae0 {
    encapsulation (ethernet-ccc | extended-vlan-ccc | vlan-ccc);
    vlan-tagging;
    aggregated-ether-options {
        minimum-links links;
        link-speed speed;
    }
    unit unit-number {
        encapsulation vlan-ccc;
        vlan-id identifier;
        family ccc;
    }
}
```

Be aware of the following limitations when configuring CCC connections over aggregated Ethernet interfaces:

If you configured load balancing between child links, be aware that a different hash key is used to distribute packets among the child links. Standard aggregated interfaces have family inet configured. An IP version 4 (IPv4) hash key (based on the Layer 3 information) is used to distribute packets among the child links. A CCC connection over an aggregated Ethernet interface has family ccc configured instead. Instead of an IPv4 hash key, a Multiprotocol Label Switching (MPLS) hash key (based on the destination media access control [MAC] address) is used to distributed packets among the child links.

The extended-vlan-ccc encapsulation is not supported on the 12-port Fast Ethernet PIC and the 48-port Fast Ethernet PIC.

The JUNOS software does not support the Link Aggregation Control Protocol (LACP) when an aggregated interface is configured as VLAN (with either the vlan-ccc or extended-vlan-ccc encapsulation). LACP can be configured only when the aggregated interface is configured with the ethernet-ccc encapsulation.

For more information about how to configure aggregated Ethernet interfaces, see the *JUNOS Network Interfaces and Class of Service Configuration Guide*.

Frame Relay Encapsulation for Layer 2 Switching Cross-Connects

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 from 1 through 511. For CCC interfaces, it must be from 512 through 4094.

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

You can configure these statements at the following hierarchy levels:

[edit logical-routers *logical-router-name* interfaces]

[edit interfaces]

PPP and Cisco HDLC Encapsulation for Layer 2 Switching Cross-Connects

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).

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

You can configure this statement at the following hierarchy levels:

[edit logical-routers *logical-router-name* interfaces]

[edit interfaces]

Defining the CCC Connection for Layer 2 Switching Cross-Connects

To configure Layer 2 switching cross-connects, define the connection between the two circuits by including the interface-switch statement. You configure this connection on the router that is acting as the switch (Router B in Figure 27 on page 360). 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.

```
interface-switch connection-name {
    interface interface-name.unit-number;
    interface interface-name.unit-number;
}
```

You can include this statement at the following hierarchy levels:

[edit protocols connections]

[edit logical-routers *logical-router-name* protocols connections]

Configuring MPLS

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

```
interface (interface-name | all);
```

You can include this statement at the following hierarchy levels:

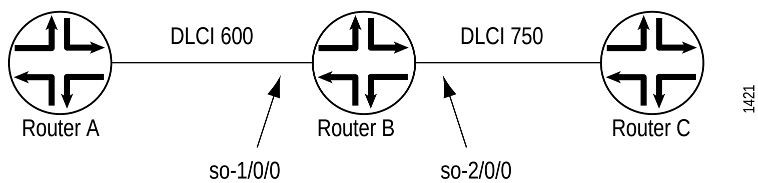
[edit protocols mpls]

[edit logical-routers *logical-router-name* protocols mpls]

Example: Configuring Layer 2 Switching Cross-Connects

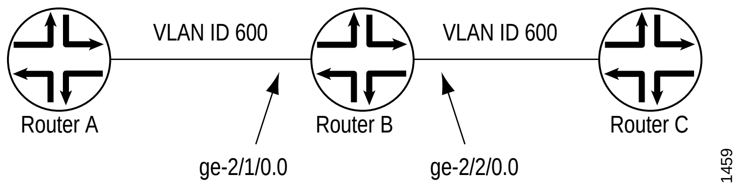
Configure a full-duplex Layer 2 switching cross-connect between Router A and Router C, using a Juniper Networks router, Router B, as the virtual switch. See the topology in Figure 28 and Figure 29 on page 368.

Figure 28: 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 29: Sample Topology of a VLAN Layer 2 Switching Cross-Connect

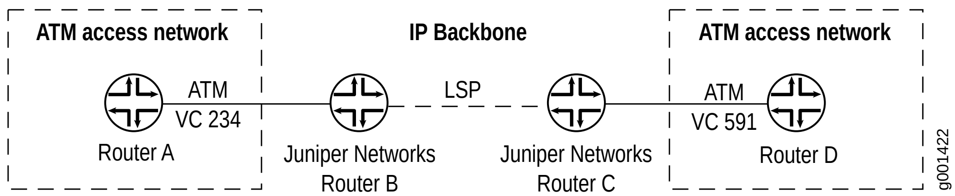



```
[edit]
interfaces {
    ge-2/1/0 {
        vlan-tagging;
        encapsulation vlan-ccc;
        unit 0 {
            encapsulation vlan-ccc;
            vlan-id 600;
        }
    }
    ge-2/2/0 {
        vlan-tagging;
        encapsulation vlan-ccc;
        unit 0 {
            encapsulation vlan-ccc;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

Configuring 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 30 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 by using an MPLS LSP.

Figure 30: 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 permanent virtual circuit (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-connect on PPP, Cisco HDLC, Frame Relay, and ATM circuits. In a single cross-connect, only like interfaces can be connected.

When you use MPLS tunnel cross-connects to support Intermediate System-to-Intermediate System (IS-IS), you must ensure that the LSP's maximum transmission unit (MTU) can, at a minimum, accommodate a 1492-octet IS-IS protocol data unit (PDU) in addition to the link-level overhead associated with the technology being connected.

For the tunnel cross-connects to work, the IS-IS frame size on the edge routers (Routers A and D in Figure 30) must be smaller than the LSP's MTU.

NOTE: Frame size values do not include the frame checksum sequence (FCS) or delimiting flags.

To determine the LSP MTU required to support IS-IS, use the following calculation:

IS-IS MTU (minimum 1492, default 1497) + frame overhead + 4 (MPLS shim header) = Minimum LSP MTU

The framing overhead varies based on the encapsulation being used. The following lists the IS-IS encapsulation overhead values for various encapsulations:

- ATM
  - AAL5 multiplex—8 bytes (RFC 1483)
  - VC multiplex—0 bytes
- Frame Relay
  - Multiprotocol—2 bytes (RFCs 1490 and 2427)
  - VC multiplex—0 bytes
- HDLC—4 bytes
- PPP—4 bytes
- VLAN—21 bytes (802.3/LLC)

For IS-IS to work over VLAN-CCC, the LSP's MTU must be at least 1513 bytes (or 1518 for 1497-byte PDUs). If you increase the size of a Fast Ethernet MTU above the default of 1500 bytes, you might need to explicitly configure jumbo frames on intervening equipment.

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 Network Interfaces and Class of Service Configuration Guide*.

To configure an LSP tunnel cross-connect, you must configure the following on the interdomain router (Router B in Figure 30 on page 369):

- Defining the CCC Encapsulation for LSP Tunnel Cross-Connects on page 371
- Defining the CCC Connection for LSP Tunnel Cross-Connects on page 372

Defining 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 30 on page 369).

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, include the encapsulation statement to configure the entire physical device. For these circuits to work, you must configure logical unit 0 on the interface.

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

You can include these statements at the following hierarchy levels:

[edit protocols interfaces]

[edit logical-routers *logical-router-name* protocols interfaces]

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

```
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 ;
    }
}
```

You can include these statements at the following hierarchy levels:

[edit protocols interfaces]

[edit logical-routers *logical-router-name* protocols interfaces]

For Frame Relay circuits, include the following statements to 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.

```
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;
    }
}
```

You can include these statements at the following hierarchy levels:

```
[edit protocols interfaces]

[edit logical-routers logical-router-name protocols interfaces]
```

For more information about the encapsulation statement, see the *JUNOS Network Interfaces and Class of Service Configuration Guide*.

Defining the CCC Connection for LSP Tunnel Cross-Connects

To configure LSP tunnel cross-connects, include the remote-interface-switch statement to define the connection between the two circuits on the ingress and egress routers (Router B and Router C, respectively, in Figure 30 on page 369). 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.

```
remote-interface-switch connection-name {
    interface interface-name.unit-number;
    transmit-lsp label-switched-path;
    receive-lsp label-switched-path;
}
```

You can include these statements at the following hierarchy levels:

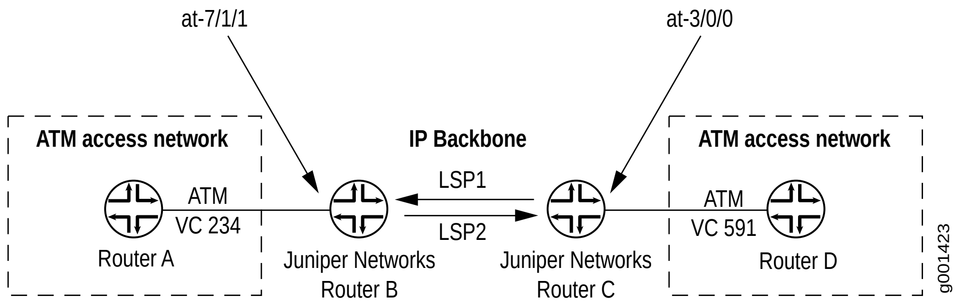
```
[edit protocols connections]

[edit logical-routers logical-router-name protocols connections]
```

Example: Configuring 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 31.

Figure 31: 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;
        }
    }
}
```

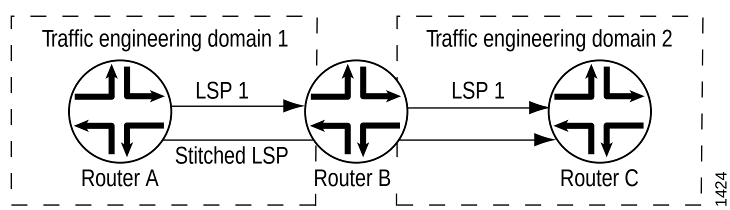
On Router C:

```
[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-3/0/0.2;
            transmit-lsp lsp2;
            receive-lsp lsp1;
        }
    }
}
```

Configuring 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 32 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 32: 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 reencapsulated 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 32), 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.

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

You can configure these statements at the following hierarchy levels:

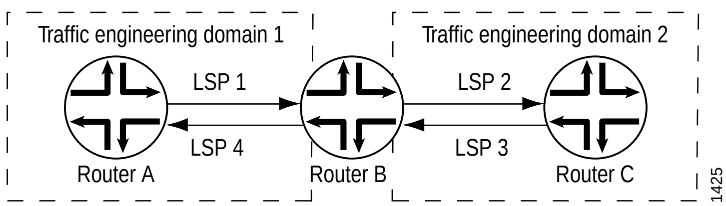
[edit logical-routers *logical-router-name* protocols connections]

[edit protocols connections]

Example: Configuring 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 33.

Figure 33: 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;
        }
    }
}
```

Transmitting Nonstandard BPDUs

CCC protocol (and Layer 2 Circuit and Layer 2 VPN) configurations can transmit nonstandard bridge protocol data units (BPDUs) generated by other vendors' equipment. This is the default behavior on all supported PICs and requires no additional configuration.

The following PICS are supported on T-series and M320 routers:

- 1-port Gigabit Ethernet PIC
- 2-port Gigabit Ethernet PIC
- 4-port Gigabit Ethernet PIC
- 10-port Gigabit Ethernet PIC

Configuring TCC

To configure translational cross-connect (TCC), you must perform the following tasks on the router that is acting as the switch:

- Defining the Encapsulation for the Layer 2 Switching TCCs on page 377
- Defining the Connection for the Layer 2 Switching TCC on page 381
- Configuring MPLS on page 382

Defining 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:

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 Encapsulations” on page 380.

PPP and Cisco HDLC Encapsulation for Layer 2 Switching TCCs

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.

To configure PPP and Cisco HDLC circuits, include the following statements:

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

You can include these statements at the following hierarchy levels:

[edit protocols interfaces]

[edit logical-routers *logical-router-name* protocols interfaces]

ATM Encapsulation for Layer 2 Switching TCCs

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:

```
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;
        }
    }
}
```

You can configure these statements at the following hierarchy levels:

[edit logical-routers *logical-router-name* protocols interfaces]

[edit protocols interfaces]

Frame Relay Encapsulation for Layer 2 Switching TCCs

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:

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

You can configure these statements at the following hierarchy levels:

[edit logical-routers *logical-router-name* protocols interfaces]

[edit protocols interfaces]

Ethernet Encapsulation for Layer 2 Switching TCCs

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:

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

You can configure these statements at the following hierarchy levels:

[edit logical-routers *logical-router-name* protocols interfaces]

[edit protocols interfaces]

For Ethernet circuits, you also need to configure the ARP. See "ARP Configuration for Ethernet TCC Encapsulations" on page 380.

Ethernet Extended VLAN Encapsulation for Layer 2 Switching TCCs

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:

```
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;
                }
            }
        }
    }
}
```

You can configure these statements at the following hierarchy levels:

[edit logical-routers *logical-router-name* protocols interfaces]

[edit protocols interfaces]

For Ethernet extended VLAN circuits, you also need to configure the ARP. See “ARP Configuration for Ethernet TCC Encapsulations” on page 380.

ARP Configuration for Ethernet TCC Encapsulations

All Ethernet TCC and Ethernet extended VLAN TCC encapsulations require that you also configure the 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 by use of the unlike Layer 2 protocol on the far side of the TCC switching router.

Configure static ARP as follows:

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

You can configure these statements at the following hierarchy levels:

[edit logical-routers *logical-router-name* protocols interfaces]

[edit protocols interfaces]

Defining the Connection for the Layer 2 Switching TCC

You must configure the connection between the two circuits of the Layer 2 switching TCC on the router acting as the switch. The connection joins the interface coming from the circuit's source to the interface leading 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.

To configure a connection for a local interface switch, include the following statements:

```
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;
}
```

You can include these statements at the following hierarchy levels:

[edit protocols connections]

[edit logical-routers *logical-router-name* protocols connections]

To configure a connection for a remote interface switch, include the following statements:

```
remote-interface-switch connection-name {
    interface interface-name.unit-number;
    interface interface-name.unit-number;
    transmit-lsp lsp-number;
    receive-lsp lsp-number;
}
```

You can include these statements at the following hierarchy levels:

[edit protocols connections]

[edit logical-routers *logical-router-name* protocols connections]

Configuring MPLS

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

To configure MPLS on an interface, include the unit statement:

unit *logical-unit-number*;

You can include this statement at the following hierarchy levels:

[edit interfaces *interface-name*]

[edit logical-routers *logical-router-name* interfaces *interface-name*]

To configure MPLS, include the interface statement:

interface (*interface-name* | all);

You can include this statement at the following hierarchy levels:

[edit protocols mpls]

[edit logical-routers *logical-router-name* protocols mpls]

Configuring CCC and TCC Graceful Restart

To enable CCC and TCC graceful restart, include the graceful-restart statement:

graceful-restart;

You can include this statement at the following hierarchy levels:

[edit routing-options]

[edit logical-routers *logical-router-name* routing-options]

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 Routing Protocols Configuration Guide*.

CCC and TCC graceful restart depend on 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 251 and “Configuring RSVP Graceful Restart” on page 271.

Configuring CCC Switching for Point-to-Multipoint LSPs

You can configure CCC to switch traffic from interfaces to point-to-multipoint LSPs. This feature is useful for handling multicast or broadcast traffic (for example, a digital video stream).

To configure CCC switching for point-to-multipoint LSPs, you do the following:

- On the ingress provider edge (PE) router, you configure CCC to switch traffic from an incoming interface to a point-to-multipoint LSP.

- On the egress PE, you configure CCC to switch traffic from an incoming point-to-multipoint LSP to an outgoing interface.

The CCC connection for point-to-multipoint LSPs is unidirectional.

For more information on point-to-multipoint LSPs, see "Point-to-Multipoint LSPs" on page 54 and "Configuring Point-to-Multipoint LSPs" on page 107.

To configure a CCC connection for a point-to-multipoint LSP, complete the steps in the following sections:

Configuring the Point-to-Multipoint LSP Switch on the Ingress PE Router

To configure the ingress PE router with a CCC switch for a point-to-multipoint LSP, configure the p2mp-transmit-switch statement.

To specify a name for the ingress CCC switch, include the p2mp-transmit-switch statement:

p2mp-transmit-switch *point-to-multipoint-transmit-switch-name*;

You can include this statement at the following hierarchy levels:

[edit protocols connections]

[edit logical-routers *logical-router-name* protocols connections]

To specify the name of the ingress interface, include the input-interface statement:

input-interface *input-interface-name.unit-number*;

You can include this statement at the following hierarchy levels:

[edit protocols connections p2mp-transmit-switch *point-to-multipoint-transmit-switch-name*]

[edit logical-routers *logical-router-name* protocols connections p2mp-transmit-switch *point-to-multipoint-transmit-switch-name*]

To specify the name of the transmitting point-to-multipoint LSP, include the transmit-p2mp-lsp statement:

transmit-p2mp-lsp *transmitting-point-to-multipoint-lsp*;

You can include this statement at the following hierarchy levels:

[edit protocols connections p2mp-transmit-switch *point-to-multipoint-transmit-switch-name*]

[edit logical-routers *logical-router-name* protocols connections p2mp-transmit-switch *point-to-multipoint-transmit-switch-name*]

Configuring the Point-to-Multipoint LSP Switch on the Egress PE Router

To configure the CCC switch for a point-to-multipoint LSP on the egress PE router, include the p2mp-receive-switch statement.

To specify a name for the egress CCC switch, include the p2mp-receive-switch statement:

p2mp-receive-switch *point-to-multipoint-transmit-switch-name*;

You can include this statement at the following hierarchy levels:

[edit protocols connections]

[edit logical-routers *logical-router-name* protocols connections]

To specify the name of the egress interface, include the output-interface statement:

output-interface *input-interface-name.unit-number*;

You can include this statement at the following hierarchy levels:

[edit protocols connections p2mp-transmit-switch *point-to-multipoint-transmit-switch-name*]

[edit logical-routers *logical-router-name* protocols connections p2mp-transmit-switch *point-to-multipoint-transmit-switch-name*],

To specify the name of the receiving point-to-multipoint LSP, include the receive-p2mp-lsp statement:

receive-p2mp-lsp *transmitting-point-to-multipoint-lsp*;

You can include this statement at the following hierarchy levels:

[edit protocols connections p2mp-transmit-switch *point-to-multipoint-transmit-switch-name*]

[edit logical-routers *logical-router-name* protocols connections p2mp-transmit-switch *point-to-multipoint-transmit-switch-name*]