

Configuring Frame Relay DLCIs

When you are using Frame Relay encapsulation on an interface, each logical interface corresponds to one or more permanent virtual circuits (PVCs) or switched virtual circuits (SVCs). For each PVC or SVC, you must configure one data-link connection identifier (DLCI).

A Frame Relay interface can be a point-to-point interface or a point-to-multipoint (also called a multipoint nonbroadcast multiaccess [NBMA]) connection.

To configure Frame Relay DLCIs, you can do the following:

- Configuring a Point-to-Point Frame Relay Connection on page 1
- Configuring a Point-to-Multipoint Frame Relay Connection on page 2
- Configuring a Multicast-Capable Frame Relay Connection on page 2

Configuring a Point-to-Point Frame Relay Connection

To configure a point-to-point Frame Relay connection, include the **dlci** statement:

```
dlci dlci-identifier;
```

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]

The DLCI identifier is a value from 16 through 1022. Numbers 1 through 15 are reserved for future use. A point-to-point interface can have one DLCI.



NOTE: For information about Frame Relay DLCI limitations for channelized interfaces, see [\[Unresolved xref\]](#).

You configure the routing platform to use DLCI sparse mode by including the **sparse-dlcis** statement at the [edit chassis fpc *slot-number* pic *pic-number*] hierarchy level. For more information about DLCI sparse mode, see the *JUNOS System Basics Configuration Guide*.

For more information about Frame Relay DLCIs, see “Configuring a Point-to-Point Frame Relay Connection” on page 1.

When you are configuring point-to-point connections, the MTU sizes on both sides of the connection must be the same.

Configuring a Point-to-Multipoint Frame Relay Connection

To configure a point-to-multipoint Frame Relay connection (also called a multipoint NBMA connection), include the `multipoint-destination` statement:

```
multipoint-destination address dlci dlci-identifier;
```

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family *family* address *address*]

For each destination, include one `multipoint-destination` statement. *address* is the address of the remote side of the connection, and *dlci-identifier* is the DLCI identifier for the connection.

When you are configuring point-to-multipoint connections, all interfaces in the subnet must use the same MTU size.

If keepalives are enabled, causing the interface to send LMI messages during idle times, the number of possible DLCI configurations is limited by the MTU selected for the interface. For more information, see *Configuring Frame Relay Keepalives*.

Configuring a Multicast-Capable Frame Relay Connection

By default, Frame Relay connections assume unicast traffic. If your Frame Relay switch performs multicast replication, you can configure the connection to support multicast traffic by including the `multicast-dlci` statement:

```
multicast-dlci dlci-identifier;
```

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]

The DLCI identifier is a value from 16 through 1022 that defines the Frame Relay DLCI over which the switch expects to receive multicast packets for replication.

You can configure multicast support only on point-to-multipoint Frame Relay connections.

If keepalives are enabled, causing the interface to send LMI messages during idle times, the number of possible DLCI configurations is limited by the MTU selected for the interface. For more information, see *Configuring Frame Relay Keepalives*.