CHAPTER 10

Configuring Frame Relay

Frame Relay was conceived as a protocol for use over serial interfaces and was designed for networks with large T1 installations. This chapter describes the tasks for configuring Frame Relay on the communication server. For a complete description of the commands mentioned in this chapter, refer to the *Communication Server Command Reference* publication. For historical background and a technical overview of Frame Relay, see the *Internetworking Technology Overview* publication.

Cisco's Implementation of Frame Relay

Cisco's Frame Relay implementation currently supports routing on IP and Novell IPX.

The Frame Relay software provides the following capabilities:

- Support for the three generally implemented specifications of Frame Relay Local Management Interfaces (LMIs):
  - — The *Frame Relay Interface* joint specification produced by Northern Telecom, Digital Equipment Corporation, StrataCom, and Cisco Systems
  - — The ANSI-adopted Frame Relay signal specification, T1.617 Annex D
  - — The CCITT-adopted Frame Relay signal specification, Q.933 Annex A
- Conformity to CCITT I-series (ISDN) recommendation as I122, "Framework for Additional Packet Mode Bearer Services."
  - — The ANSI-adopted Frame Relay encapsulation specification, T1.618
  - — The CCITT-adopted Frame Relay encapsulation specification, Q.922 Annex A
- Conformity to Internet Engineering Task Force (IETF) encapsulation in accordance with RFC 1294 (except bridging).

- Support for a keepalive mechanism, a multicast group, and a status message, as follows:
  - The keepalive mechanism provides an exchange of information between the network server and the switch to verify that data is flowing.
  - The multicast mechanism provides the network server with its local data link connection identifier (DLCI) and the multicast DLCI. This feature is specific to our implementation of the Frame Relay joint specification.
  - The status mechanism provides an ongoing status report on the DLCIs known by the switch.
- Transmission of congestion information from Frame Relay to DECnet Phase IV and CLNS. This mechanism promotes Forward Explicit Congestion Notification (FECN) bits from the Frame Relay layer to upper-layer protocols after checking for the FECN bit on the incoming DLCI. Use this Frame Relay congestion information to adjust the sending rates of end hosts. FECN-bit promotion is enabled by default on any interface using Frame Relay encapsulation. No configuration is required.
- Support for Frame Relay Inverse Address Resolution Protocol (InvARP) as described in RFC 1293 for the IP and IP protocols. It allows a communication server running Frame Relay to discover the protocol address of a device associated with the virtual circuit.
- Support for Frame Relay switching, whereby packets are switched based on the DLCI (a Frame Relay equivalent of a MAC-level address). Communication servers are configured as a hybrid DTE switch or pure Frame Relay DCE access node in the Frame Relay network. Cisco's implementation of Frame Relay switching allows the following configurations:
- Support for *subinterfaces* associated with a physical interface. The software groups one or more permanent virtual circuits under separate subinterfaces, which in turn are located under a single physical interface. See Chapter 6, "Configuring Interfaces," for more information. See the sections Associate a DLCI with a Subinterface and Frame Relay Configuration Examples later in this chapter for more information about subinterfaces in Frame Relay configurations.
- Support of the Frame Relay DTE MIB specified in RFC 1315. However, the error table is not implemented. To use the Frame-Relay MIB, refer to your MIB publications.

Frame Relay Hardware Requirements

One of the following hardware configurations is possible for Frame Relay connections:

- Communication servers can connect directly to the Frame Relay switch.
- Communication server can connect directly to a Channel Service Unit/Digital Service Unit (CSU/DSU) first, and the CSU/DSU is connected to a remote Frame Relay switch.

Note A Frame Relay network is not required to support only communication servers that are connected directly or only communication servers connected via CSU/DSUs. Within a network, some communication servers can connect to a Frame Relay switch through a direct connection and others through connections via CSU/DSUs. However, a single communication server interface configured for Frame Relay can be only one or the other.

The Frame Relay interface actually consists of one physical connection between the network server and the switch that provides the service. This single physical connection provides direct connectivity to each device on a network, such as a StrataCom FastPacket wide area network.

Frame Relay Configuration Task List

There are required, basic steps you must follow to enable Frame Relay for your network (see the Enable Frame Relay on an Interface section). In addition, you can customize Frame Relay for your particular network needs, set local and multicast DLCIs in test environments, and monitor Frame Relay connections. These tasks are outlined next.

- Enable Frame Relay on an Interface
- Customize Your Frame Relay Network
- Configure Frame Relay in a Test Environment
- Monitor the Frame Relay Connections

The following sections describe these tasks. See the examples at the end of this chapter for ideas of how to configure Frame Relay on your network. See the *Communication Server Command Reference* for information about the commands listed in the tasks.

Enable Frame Relay on an Interface

You must perform the following tasks to enable Frame Relay:

- Set Frame Relay encapsulation
- Establish mapping

Set Frame Relay Encapsulation

To set Frame Relay encapsulation at the interface level, perform the following task in interface configuration mode:

Task	Command
Enable Frame Relay and specify the encapsulation method.	**encapsulation frame-relay** [**ietf**]

Frame Relay supports encapsulation of all supported protocols in conformance with RFC 1294, allowing interoperability between multiple vendors. Use the IETF form of Frame Relay encapsulation if your communication server is connected to another vendor's equipment across a Frame Relay network. IETF encapsulation is supported at either the interface level or on a per-DLCI (map entry) basis.

For an example of how to enable Frame Relay and set the encapsulation method, see the section "Example of Configurations using IETF Encapsulation" and Example of Two Communication Servers in Static Mode later in this chapter.

Establish Mapping

The Frame Relay map tells the network server how to get from a specific protocol and address pair to the correct local data link connection identifier (DLCI). To establish mapping according to your network needs, perform one of the following tasks in interface configuration mode:

Task	Command
Define the mapping between a supported protocol address and the DLCI used to connect to the address.	**frame-relay map** *protocol protocol-address DLCI* [**broadcast**] [**ietf**] [**cisco**]

The supported protocols with the corresponding keywords to enable them are as follows:

- IP— **ip**
- Novell IPX—**ipx**

This command is not required if you are using inverse ARP.

The configuration for the Open Shortest Path First (OSPF) protocol can be greatly simplified by adding the optional **broadcast** keyword when doing this task. See the **frame-relay map** description in the *Communication Server Command Reference* publication and the examples at the end of this chapter for more information about using the **broadcast** keyword.

For an example of how to establish mapping, see the section Example of Two Communication Servers in Static Mode and Example of Routing Novell Packets later in this chapter.

Customize Your Frame Relay Network

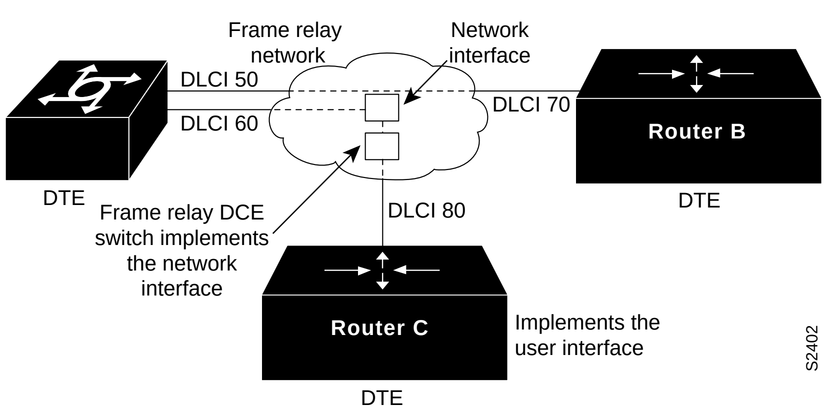
Perform the following tasks to customize Frame Relay:

- Configure Frame Relay switching
- Configure the LMI
- Select Frame Relay Inverse ARP
- Associate a DLCI with a subinterface
- Define subinterfaces

Configure Frame Relay Switching

Frame Relay switching is a means of switching packets based upon the DLCI, which can be looked upon as the Frame Relay equivalent of a MAC address. The switching is performed by configuring your communication server as a Frame Relay network. There are two parts to a Frame Relay network: a Frame Relay DTE (the communication server) and a Frame Relay DCE switch. Figure 1-1 illustrates this concept.

Figure 1-1 Frame Relay Switched Network



In Figure 1-1, The communication server and routers A, B, and C are Frame Relay DTEs connected to each other via a Frame Relay network. Our implementation of Frame Relay switching allows our communication servers to be used as depicted in this Frame Relay network.

Perform these tasks, as necessary, to configure Frame Relay switching:

- Enable Frame Relay switching
- Configure a Frame Relay DTE device, DCE switch, or NNI support
- Specify the static route

These tasks are described in the following sections.

Enable Frame Relay Switching

You must enable packet switching before you can configure it on a Frame Relay DTE, DCE, or with Network to Network Interface (NNI) support. Do this by performing the following task in global configuration mode before configuring the switch type:

Task	Command
Enable Frame Relay switching.	**frame-relay switching**

For an example of how to enable Frame Relay switching, see the switching example later in this chapter.

Configure a Frame Relay DTE Device, DCE Switch, or NNI Support

You can configure your communication server as a DTE device, DCE switch, or as a switch connected to a switch (to support NNI connections). To do so, perform the following task in interface configuration mode:

Task	Command
Configure a Frame Relay DTE device or DCE switch.	**frame-relay intf-type** [**DTE** | **DCE** | **NNI**]

Use the **DTE** keyword to configure a DTE device. DTE is the default. Use the **DCE** keyword to configure a DCE switch. Use the **NNI** keyword with this task to configure NNI support.

Specify the Static Route

You must specify a static route for PVC switching. To do so, perform the following task in interface configuration mode:

Task	Command
Specify the static route for PVC switching.	**frame-relay route** *in-DLCI out-interface out-DLCI*

Configure the LMI

Our Frame Relay software supports the industry-accepted standards for addressing the Local Management Interface (LMI), including the Cisco specification. You can enable the following LMI features:

- Set the LMI type, either ANSI, CCITT, or Cisco.
- Set the LMI keepalive interval.
- Set LMI polling intervals, timer intervals, and error thresholds,; parameters exist for both DTE and DCE device types.

Set the LMI Type

You can set one of three types of LMIs on a Cisco communication server: ANSI T1.617 Annex D, CCITT, and Cisco. To do so, perform the following task in interface configuration mode:

Task	Command
Set the LMI type.	f**rame-relay lmi-type** {**cisco** | **ansi** | **ccitt**}

Set the LMI Keepalive Interval

A keepalive interval must be set to enable LMI. By default, this interval is ten seconds and, per the LMI protocol, must be less than the corresponding interval on the switch. To do so, perform the following task in interface configuration mode:

Task	Command
Set the keepalive interval	f**rame-relay keepalive** *number*
Turn off keepalives on networks without an LMI.	**no frame-relay keepalive**

This command has the same effect as the **keepalive** interface configuration command.

The keepalive interval cannot be enabled when the LMI is disabled; they go together. For an example of how to specify an LMI keepalive interval, see the section Example of Two Communication Servers in Static Mode later in this chapter.

Set the LMI Polling and Timer Intervals

You can set various counters, intervals, and thresholds to fine-tune the operation of your LMI DTE and DCE devices. See the following table for the tasks that you can perform. See the *Communication Server Command Reference* publication for details about commands used to set the polling and timing intervals. Set these intervals, by performing one or more of the following tasks in interface configuration mode:

Task	Command
Set the DCE and NNI error threshold.	**frame-relay lmi-n392dce** *threshold*
Set the DCE and NNI monitored events count.	f**rame-relay lmi-n393dce** *events*
Set the polling verification timer on a DCE or NNI interface.	**frame-relay lmi-t392dce** *timer*
Set a full status polling interval on a DTE or NNI interface.	f**rame-relay lmi-n391dte** *keep-exchanges*
Set the DTE or NNI error threshold.	**frame-relay lmi-n392dte** *threshold*
Set the DTE and NNI monitored events count.	**frame-relay lmi-n393dte** *events*

Select Frame Relay Inverse ARP

You can select Frame Relay Inverse ARP as a method of building dynamic routes in Frame Relay networks running IP and Novell IPX. Inverse ARP allows the communication server to discover the protocol address of a device associated with the virtual circuit. Inverse ARP is used instead of the **frame-relay map** command which allows you to define the mappings between a specific protocol and address and a specific DLCI (see the section Establish Mapping earlier in this chapter for more information).

Usually you would only select Inverse ARP if you want to configure an interface for multipoint communication. You would not need to select Inverse ARP if you have a point-to-point interface because there is only a single destination and discovery is not required.

To select Inverse ARP, perform the following task in interface configuration mode:

Task	Command
Select Frame Relay Inverse ARP.	**frame-relay inverse-arp** *protocol DLCI*

Inverse ARP is enabled by default.

Define Subinterfaces

Subinterfaces solve many of the problems seen in protocols that have split horizon enabled and no capability to disable it. However, not all protocols support subinterfaces. See the“Configuring Interfaces” chapter in this guide for information for a list of protocols that support subinterfaces.

You can configure subinterfaces for multipoint or point-to-point communication. Point-to-point is the default. To configure an interface for multipoint or point-to-point communication, you must first define an interface in global configuration mode. After defining an interface, you can define a subinterface for that interface by performing the following task in interface configuration mode:

Task	Command
Define a subinterface.	**interface** *interface-type subinterface-number* [**multipoint** | **point-to-point**]

See the “Configuring Interface” chapter for more information about the **interface** command and subinterfaces.

Once you have defined the subinterface, you must perform one of the tasks in interface configuration mode:

- Establish mapping using the **frame-relay map** command.
- Select Frame-Relay Inverse ARP using the **frame-relay inverse-arp** command.
- Associate a DLCI with a subinterface using the **frame-relay interface-dlci** command.

If you define a subinterface for multipoint communication, you cannot use the **frame-relay-interface-dlci** command. If you define a subinterface for point-to-point communication, you cannot use the **frame-relay map** command. The **frame-relay inverse-arp** command is designed for use with an interface configured for multipoint communication and should not be used for a subinterface configured for point-to-point communication.

Note If you define a subinterface for point-to-point communication, you cannot reassign the same subinterface number to be used for multipoint communication without first rebooting the communication server.

Associate a DLCI with a Subinterface

You must associate the Frame Relay DLCI with a subinterface to use subinterfaces in the Frame Relay network for point-to-point communication. If you associate a DLCI with a point-to-point subinterface, you cannot use the **frame-relay map** command. To do so, perform the following task in interface configuration mode:

Task	Command
Associate a DLCI with a subinterface.	**frame-relay interface-dlci** *DLCI* [*options*]

Configure Frame Relay in a Test Environment

Perform the following tasks only if you are configuring Frame Relay in a test environment:

- Set the local DLCI
- Set the DLCI for multicasts

Set the Local DLCI

You can set a local DLCI in a test environment. This feature is provided mainly to allow testing of the Frame Relay encapsulation in a setting where two communication servers are connected back to back. This command is not required in a live Frame Relay network. Its use allows the source local DLCI to be set for use when the LMI is not supported. To do so, perform the following task in interface configuration mode:

Task	Command
Set a local DLCI.	**frame-relay local-dlci** *number*

If LMI is supported and the multicast information element is present, the network server sets its local DLCI based on information provided via the LMI.

Set the DLCI for Multicasts

You can specify a DLCI for multicasts in a test environment. This feature is provided mainly to allow testing of the Frame Relay encapsulation in a setting where two communication servers are connected back to back. This task is not required in a live Frame Relay network. Its use allows network transmissions (packets) sent to a multicast DLCI to be delivered to all network servers defined as members of the multicast group. To set the DLCI for multicasts, perform the following task in interface configuration mode:

Task	Command
Specify a DLCI for multicasts in a test environment.	**frame-relay multicast-dlci** *number*

Monitor the Frame Relay Connections

To monitor Frame Relay connections, perform any of the following tasks in interface configuration mode:

Task	Command
Display information about the Frame Relay DLCI and the LMI.	**show interfaces serial** *unit*
Display LMI statistics.	**show frame-relay lmi** [*interface*]
Display the current Frame Relay map entries.	**show frame-relay map**
Display PVC statistics.	**show frame-relay pvc** [*interface* [*DLCI*]]
Display configured static routes.	**show frame-relay route**
Display Frame Relay traffic statistics.	**show frame-relay traffic**

Frame Relay Configuration Examples

This section provides examples of Frame Relay configurations. The following examples are provided:

- Example of Configurations using IETF (page 10-9)
- Example of Two communication servers in static mode (page 10-10)
- Example of routing Novell packets (page 10-10)
- Example of configuration providing backward compatibility (page 10-10)
- Example of netbooting over Frame Relay (page 10-11)
- Example of switching over an IP tunnel (page 10-12)

Example of Configurations Using IETF Encapsulation

The first example that follows sets IETF encapsulation at the interface level. The second example sets IETF encapsulation on a per-DLCI basis. In the first example, the keyword **ietf** sets the default encapsulation method for all maps to IETF.

```
encapsulation frame-relay IETF
frame-relay map ip 131.108.123.2 48 broadcast
frame-relay map ip 131.108.123.3 49 broadcast
```

In the following example, IETF encapsulation is configured on a per-DLCI basis. This configuration has the same result as the configuration in the first example.

```
encapsulation frame-relay
frame-relay map ip 131.108.123.2 48 broadcast ietf
frame-relay map ip 131.108.123.3 49 broadcast ietf
```

Example of Two Communication Servers in Static Mode

The following examples illustrate how to configure two communication servers for static mode.

Configuration for Communication Server 1

```
interface  serial 0
!
ip address  131.108.64.2 255.255.255.0
encapsulation  frame-relay
keepalive 10
frame-relay map ip  131.108.64.1 43
```

Configuration for Communication Server 2

```
interface serial 0
!
ip address 131.108.64.1 255.255.255.0
encapsulation  frame-relay
keepalive 10
frame-relay map ip  131.108.64.2 44
```

Example of Routing Novell Packets

The following example illustrates how to send packets destined for Novell address 200.0000.0c00.7b21 out on DLCI 102.

```
interface ethernet 0
ipx network 2abc
!
interface serial 0
ipx network 200
encapsulation frame-relay
frame-relay map ipx 200.0000.0c00.7b21 102 broadcast
!
```

Example of a Configuration Providing Backward Compatibility

The following configuration provides backward compatibility and interoperability. These functions are possible by using the flexibility provided by separately defining each map entry.

```
encapsulation frame-relay
frame-relay map ip 131.108.123.2 48 broadcast ietf
! interoperability is provided by IETF encapsulation
frame-relay map ip 131.108.123.3 49 broadcast ietf
frame-relay map ip 131.108.123.7 58 broadcast
! this line allows the communication server to connect with a
! device running an older version of software
frame-relay map ipx 21.7 49 broadcast
```

Configure IETF based on map entries and protocol for more flexibility. Use this method of configuration for backward compatibility and interoperability.

Example of Netbooting over Frame Relay

When netbooting over Frame Relay, you cannot netboot via a broadcast. You must netboot from a specific host. Also, a **frame-relay map** command must exist for the host that you will netboot from.

For example, if file gs3-bfx is to be booted from a host with IP address 131.108.126.2, the following commands would need to be in the configuration:

```
boot system gs3-bfx 131.108.126.2

interface Serial 0
encapsulation frame-relay
frame-relay map IP 131.108.126.2 100 broadcast
```

The **frame-relay map** command is used to map an IP address into a DLCI address. In order to netboot over Frame Relay, the address of the machine to netboot from must be given explicitly, and a **frame-relay map** entry must exist for that site. For example:

```
boot system gs3-bfx.83-2.0 131.108.13.111
!
interface Serial 0
ip address 131.108.126.200 255.255.255.0
encapsulation frame-relay
!
frame-relay map IP 131.108.126.111 100 broadcast
```

In this case, 100 is the DLCI of the remote communication server than can get to host 131.108.13.111.

The remote communication server must have the following **frame-relay map** entry:

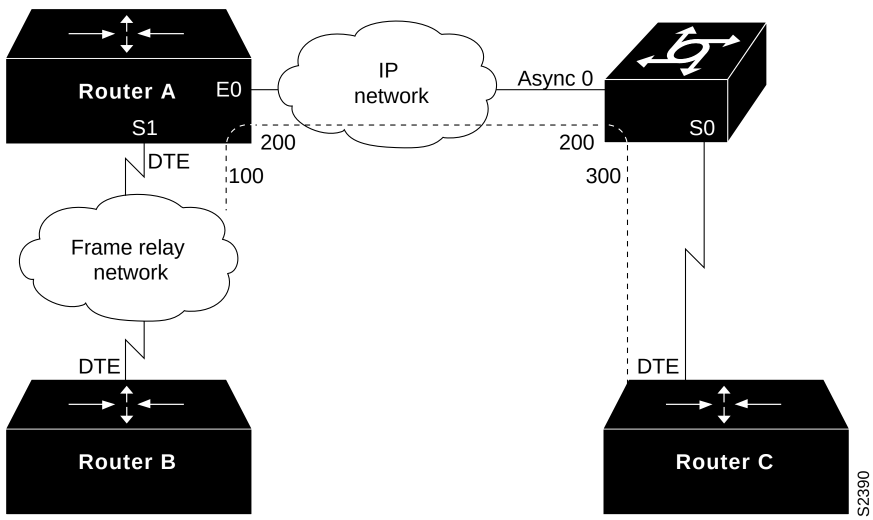
```
frame-relay map IP 131.108.126.200 101 broadcast
```

This entry allows the remote communication server to return a boot image (from the netboot host) to the communication server netbooting over Frame Relay. Here, 101 is the DLCI of the communication server being netbooted.

Example of Switching over an IP Tunnel

Switching over an IP tunnel is done by creating a point-to-point tunnel across the internet over which PVC switching can take place (see Figure 1-2).

Figure 1-2 Frame Relay Switch over IP Tunnel



The following example illustrates how to create the IP network depicted in Figure 1-2.

Configuration for Router A

```
frame-relay switching
!
interface Ethernet0
ip address 108.131.123.231 255.255.255.0
!
!
interface Serial0
no ip address
ip address 131.108.222.231 255.255.255.0
encapsulation frame-relay
frame-relay map ip 131.108.222.4 400 broadcast
frame-relay route 100 interface Tunnel1 200
!
interface Tunnel1
tunnel source Ethernet0
tunnel destination 150.150.150.123
```

Configuration for the Communication Server

```
frame-relay switching
!
interface Async0
ip address 131.108.231.123 255.255.255.0
encapsulation ppp
!
interface Serial0
ip address 150.150.150.123 255.255.255.0
encapsulation ppp
!
interface Tunnel1
tunnel source Async0
tunnel destination 108.131.123.231
!
interface Serial1
ip address 131.108.7.123 255.255.255.0
encapsulation frame-relay
frame-relay intf-type dce
frame-relay route 300 interface Tunnel1 200
```